Comparison of High-Order Methods on Unstructured Grids

Kashif H. Iqbal

Submitted for the Degree of Ph.D.

Fluid Mechanics and Computational Science Group
School of Engineering
Cranfield University
Cranfield, UK

2013

Abstract

A high-order Discontinuous Galerkin (DG) method is formulated and implemented on the Cranfield University's 3D unstructured Finite Volume Method (FVM) code (UCNS3D), for both linear and non-linear hyperbolic conservation laws and for test-cases which exhibit both smooth and discontinuous solutions . As both DG and FVM are developed on the same solver platform, this enables the use of any procedures which are common to both the methods, thus, ensuring the closest possible comparison.

The initial part of the thesis details the basic concepts and derivation of the discontinuous Galerkin method in the 1D space for the advection equation, which is then extended to the 3D space for a hyperbolic system.

Prior to comparing the FVM and DG methods, the DG method implementation is verified . The verification is a combination of a theoretical and numerical approach which endeavours to minimize any potential programming errors.

Following the verification of the DG method, the FVM and DG methods are compared for numerous flows: the linear advection equation and Euler equations, sufficiently smooth testcases, and testcases which require a limiter to suppress Gibb's oscillations.

Acknowledgements

Firstly I would like to thank my supervisor Evgeniy Shapiro who left the department before I could submit this thesis, without whose input, encouragement and drive this PhD would not have been completed. I would also like to thank Professor Dimitris Drikakis for allowing me the opportunity to conduct research in his group.

A thank you to members of office F51 Tareq, Baba, Francis, Samer, Peter, Yosuka, Hakim, Hamid, Mudassir, Neal and Leo and for their camaraderie. In particular I'd like to mention: Ramey Jamil and Guido Monterzino for their late night conversations, Antonis Antoniadis for his help with the Finite Volume Method, Solange Baena and Loraine Thomas for their friendship.

A special thank you to my girlfriend, Rabia, who had to endure the unsociable hours and missed dates.

Finally, I would want to pay tribute to my family who have always encouraged me to better myself, My mother Rehana, father Pervez and brothers and sisters, Hussain, Kazim, Rahat and Aimah.

Contents

Nomenclature

LATIN

a_l	-	Norm of the basis function
C_x, C_y, C_z	-	Advection velocity: x,y,z respectively
F, G, H	-	Conservative flux
$\hat{F}, \hat{G}, \hat{H}$	-	Numerical flux
$\mathbf{F}$	-	Vector of conservative fluxes
$\hat{\mathbf{F}}$	-	Vector of numerical fluxes
$\ell_l(x)$	-	Lagrangian Polynomials
L_∞,L_1	-	Error norms
$\mathbf{L_h}$	-	Right-hand side of weak formulation
$\mathbf{M}$	-	Mass matrix
N_{DOF}	-	Total number of degrees of freedom
N_E	-	Number of elements
N_G	-	Number of grid points
N_p	-	Number of degrees of freedom
$\hat{\mathbf{n}}$	-	Outward unit normal
P_l	-	Polynomial order
$P_l^{\alpha,\beta}$	-	Jacobi Polynomial

RK	-	Order of Runge Kutta Method
S_I	-	Surface Integral
$\|S_F\|$	-	Surface area of face F of tetrahedron
$\bar{U}$	-	Cell average
U_h	-	Approximate solution
U_h^L	-	Approximate solution for left state
U_h^R	-	Approximate solution for the right state
$\hat{u}_l(t)$	-	Degree of freedom
$\bar{U}^{max}$	-	Maximum cell averages of the neighbours
$\bar{U}^{min}$	-	Minimum cell averages of the neighbours
x, y, z	-	Physical coordinates
x_i	-	Midpoint of element i in 1D DG
$\|V\|$	-	Volume of physical tetrahedron
V_I	-	Volume Integral

GREEK

Δ_i	-	The element size in 1D DG
ϕ_l	-	Basis function
Ω	-	Physical domain
$\hat{\Omega}$	-	Reference domain
$\partial\Omega$	-	Surface in physical domain
$\partial\hat{\Omega}$	-	Surface in reference domain

ξ, η, ζ	-	Reference coordinates
$\chi_{x\xi}$	-	Mapping from the physical to the reference space
ξ_p^S, η_p^S	-	Points defined on the reference riangle
$\xi_p^V, \eta_p^V, \zeta_p^V$	-	Points defined on the ference tetrahedron
ψ_j	-	Flux limiter function at quadrature point j

ABBREVIATIONS

CFL	-	Courant-Friedrichs-Lewy Number
DG	-	Discontinuous Galerkin
DG2	-	2nd Order Discontinuous Galerkin
DG3	-	3rd Order Discontinuous Galerkin
ENO	-	Essentially Non-Oscillatory
FDM	-	Finite Difference Method
FEM	-	Finite Element Method
FVM	-	Finite Volume Method
FVM2	-	2nd Order Finite Volume Method
FVM3	-	3rd Order Finite Volume Method
HLLC	-	Harten-Lax-van Leer-Contact Solver
ODE	-	Ordinary Differential Equation
PDE	-	Partial Differential Equation
TVB	-	Total Variation Bounded
TVD	-	Total Variation Diminishing

WENO	-	Weighted Essentially Non-Oscillatory

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation

With the advances in computing technology, with Gordon Moore predicting the number of transistors on a CPU will double every 2 years (**Figure 1.2**), something which has been observed in practice, and the dropping cost of computers, the benefits of using simulation based research has been becoming more attractive. **Figure 1.1** shows the rapid increase in the computing power over an 8 year period, at present (June 2013) the world's fastest supercomputer, *Tianhe-2*, at the National University of Defence Technology in China, achieved a performance of 33.86 Petaflops. When compared with 8 years prior, the fastest supercomputer in the world, BlueGene/L, achieved a performance of 101.4 Teraflops, representing an increase of approximately 300 times the computing power. This increase in computing power has made it possible to conduct in depth research in computationally demanding areas, some of the areas in which modelling and simulation have become an indispensable tool, including: molecular dynamics, fluids dynamics, solid mechanics and nuclear weapons testing.

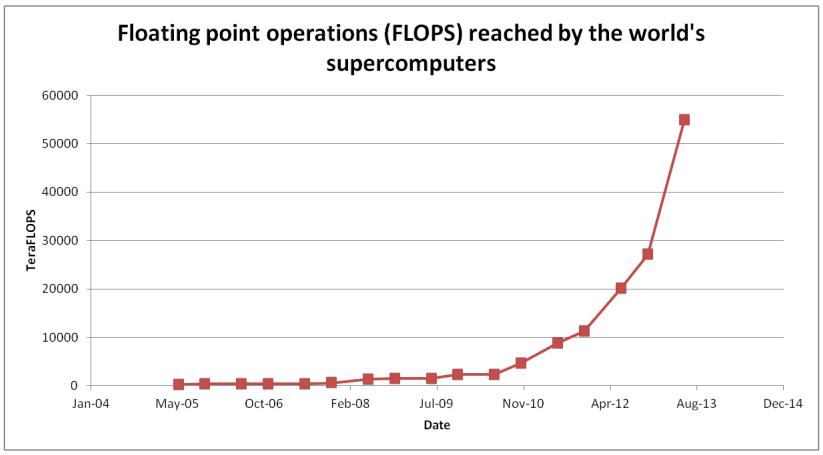


Figure 1.1: Trend in supercomputer performance over 8 years

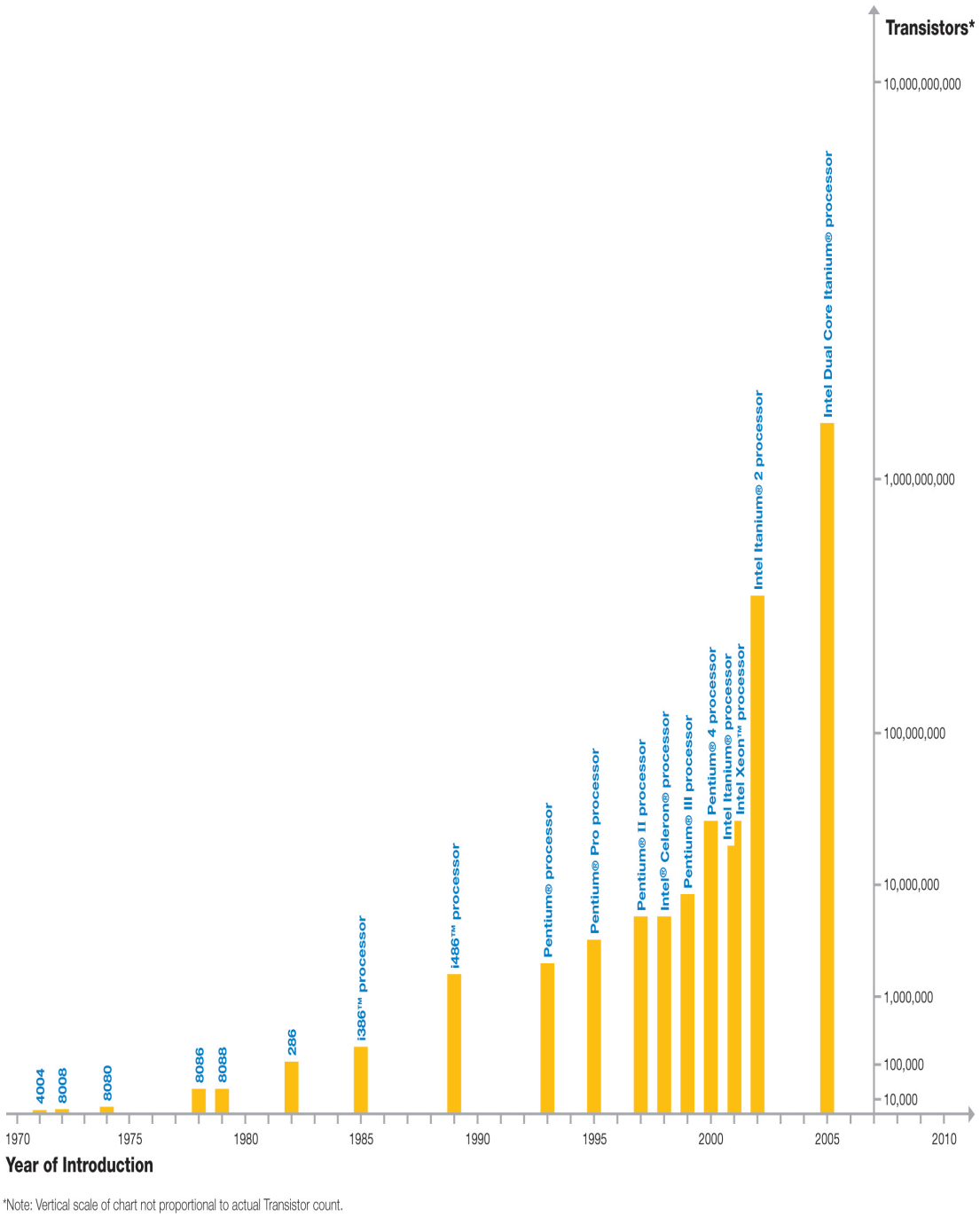


Figure 1.2: Increase in number of transistors on CPUs over 35 years

Inevitably the increase in power and the decline in cost of computing hardware has popularised scientific modelling. Although another aspect of scientific modelling has to be consider - *numerical methods*. Numerical methods can be viewed as the foundation of scientific computing, upon which the accuracy, speed of execution and other aspects of the simulation rest. To date, many numerical methods have been proposed, with each one displaying particular strengths and weaknesses. Concentrating on Computational Fluid Dynamics (CFD), the Finite Volume Method (FVM) is seen as the numerical method of choice. FVM has been used extensively in academia and industry, and the majority of commercial off-the-shelf solvers (including *Fluent* and *STAR-CCM+*) and the open source project *OpenFOAM* utilise this method.

As will be shown in the proceeding chapters, within the framework of FVM high-resolution methods have been developed, which have demonstrated the simulation of complicated flows such as turbulence and multiphase flows. However, the greatest drawback of FVM is the disproportionately large stencils required to extend the order of accuracy in 3D, which is subsequently made more difficult on an unstructured grid. To overcome this problem researchers have developed different methods or modified existing ones. Since its development 20 years ago the Discontinuous Galerkin (DG) method has become very popular with researchers, which retains the positive aspects other numerical methods but the circumvention of their drawbacks (**Table 1.1**).

Table 1.1: Comparison of numerical methods, replicated from[28]

	Complex geometries	h/p adaptivity	Explicit form	Conservation law	Elliptic problems
FDM	×	✓	✓	✓	✓
FVM	✓	×	✓	✓	(✓)
FEM	✓	✓	×	(✓)	✓
DG	✓	✓	✓	✓	(✓)

✓ - the method has been succesfully applied or is naturally suited for this particular feature

(✓) - the method can be applied with some modification, but it is not a natural choice

× - the method is very difficult or impossible to apply to such problems

FVM and DG share common features which makes the comparison and implementation of both methods on a single solver platform possible. Thus, this poses the natural question: is the DG method better, worse or comparable in accuracy, speed of execution and implementation to the more established and widely used FVM?

1.2 Numerical Methods

Initially, the techniques developed in CFD were done with respect to the Finite Difference Method (FDM)[72]. However, this technique lost favour with the CFD community when it became apparent that FDM could not easily be applied to irregular

grids, namely unstructured grids. Another issue was the inability of FDM to preserve the conservative nature of the governing equations[43]. This has not hampered the use of FDM though, with three-dimensional simulations being carried out which show excellent shock-capturing abilities[53], and the development of high-resolution methods[60].

The need for unstructured grids is necessitated by the complicated geometries associated with the simulation of complex problems. The issue of unstructured grids is overcome by the use of the FVM and the Finite Element Method(FEM). In the Finite Element Method (FEM), the use of a large stencil is eliminated and the accuracy is governed by the polynomial approximation which is determined within the element. In the case of a compressible flows, the particular type of FEM which is used is the Streamline Upwind Petrov-Galerkin Method (SUPG) proposed by *Brooks and Hughes* [6]. In SUPG a perturbation term is added to the weighting function of the standard Galerkin formulation, the purpose of which is to minimise the artificial diffusion in the flow direction[65]. If explicit time discretisation is used then a couple systems has to be solved at each time-step, this is because the DOF are shared between the elements.

1.2.1 Finite Volume Method

The Finite Volume Method (FVM) become extremely popular in the development and use of CFD because of its ability to handle unstructured grids (something which is difficult to do with FDM as mentioned previously). As a result many innovations in CFD have been developed in the context of FVM, or with FVM in mind. As there exists a discontinuity (mathematically speaking) between the cells a *Riemann problem* usually has to be solved between each cell face, which gives the FVM the ability to effectively model shocks[17, 38]. Alternatively, *Jameson, Schmidt and Turkel* developed the *JST* or *Jameson scheme* [30] which resolves the flux integral without the need for a Riemann solver.

Godunov's method of solving PDEs by assuming the conserved variables as piecewise constant values made it possible to solve hyperbolic conservation laws on unstructured grids, although this method is 1st order accurate it laid the foundations for further higher-order attempts. The first high-order attempt for FVM was made by *Van Leer*[39] - the Monotone Upstream-centred schemes for Conservation Laws (MUSCL) - in which the MUSCL scheme is 2nd order in both space and time[1].

The idea behind the MUSCL scheme was to replace the piecewise constant of Godunov's method by a piecewise linear approximations and limit the slope of the linear approximation so that they don't exhibit oscillations. The MUSCL scheme has been extended to 3rd order accuracy[15] and 5th order accuracy[63]. The MUSCL scheme is part of a wider class of schemes called total variation diminishing (TVD), a concept proposed by *Harten*[26]. TVD schemes ensures, as time increases, the total variation

[1] It has been noticed that Kolgan, may have successfully developed the first TVD scheme in the context of FDM[37]

of the reconstructed data is either equal to, or diminishes with respect to the initial data. TVD schemes have been proven to be monotonicity preserving, and as a result no oscillations are introduced into the method for high orders of accuracy. The idea of TVD is not unique to FVM, and is a property of a numerical method, therefore, other numerical methods can be TVD. The central idea of a TVD scheme is a limiter which suppresses oscillations, there are a variety of limiters available[64], unfortunately, there is no way of knowing beforehand which limiter will provide the best resolution. TVD schemes can introduce too much numerical diffusion into the problem, and become unreliable for problems which have flow structures which require a large amount of time to develop[49]. To further add to the woes of excessive numerical diffusion of the TVD scheme, *Goodman and LeVeque*[22] proved that any TVD method in 2D or 3D are at most 1st order accurate[2]. However, in practice TVD schemes have been successfully applied to 2D or 3D problems. A review of TVD methods is given in[68].

A further class of methods have been developed which improve on the TVD schemes, the Essentially Non-Oscillatory (ENO) of *Harten, Engquist, Osher and Chakravarthy* [27] and Weighted Essentially Non-Oscillatory (WENO) schemes due to *Lui,Osher and Chan*[46]. ENO scheme operate on the concept of choosing the smoothest possible stencil, given r possible stencils. The chosen stencil are then used to construct the reconstruction polynomial. The WENO approach modifies the ENO scheme by using a convex combination of all the possible stencils. It is vital to remember that (W)ENO schemes are not TVD, they will introduce bounded oscillations into the problem[38], however, these will decrease as the grid is refined[7]

The TVD and (W)ENO are collectively known as high resolution methods,and the library of problems to which high resolution methods have been applied is extensive, including: incompressible flows, compressible flows, hypersonic flows and magentohydrodynamics [7, 17, 32, 35], to name but a few.

Unfortunately, to increase the accuracy of FVM in 3D the stencil has to be extended to such an extent that it becomes disproportionately large. Unlike in 1D where the stencil increases at a linear rate with respect to reconstruction polynomial, in 2D and 3D the increase in the stencil size is proportional to r^2 and r^3 respectively. For the stencil(in 3D), the number of points of the stencil is:

$$K = \frac{1}{6}(r+1)(r+2)(r+3) - 1$$

Where K is the minimum number of elements comprising a single stencil, and r is the order of the reconstruction polynomial. Therefore a single 3rd order(P_2) stencil in 3D requires a minimum of 9 elements. In addition, for high-order schemes more than one Riemann problem is solved per face, therefore, more than one stencil reconstruction is required. It has been noted by[52] that for robustness to use between 50-100% more elements. In[18] total of $M \approx 2K$ stencils were employed for a central stencil reconstruction.

[2] except for in trivial cases

1.2.2 Discontinuous Galerkin

A relatively new method to gain attention within the computational physics' community is Runge-Kuuta Discontinuous Galerkin (RKDG). The spatial discretisation is preformed using Discontinuous Galerkin - closely resembling a Galerkin method which is extensively used in Finite Element Methods (FEM). The temporal discretisation is preformed using a Runge-Kutta method. The RKDG method encompasses many useful properties, the ability to model discontinuities which are characteristics features of compressible flows, whilst retaining a very high order of accuracy. Due to the aforementioned advantages, there has been an increasing interest in the application of RKDG over the last two decades.

The genesis of RKDG can be traced back to Reed and Hill[57], who in their 1973 paper discretised the domain using DG to solve a steady-state neutron-transport equation. It is vital to note the equation solved by Reed and Hill did not require a time-stepping method. LeSaint and Raviart carried out the first analysis of Reed and Hill's method and proved the rate of convergence for Reed and Hill's investigation. Numerous authors contributed to the early theoretical development of DG, and a brief history is given in [2].

Since the publication of Reed and Hill's seminal paper, extensive research has been carried out on further developing DG. The natural direction is the application to the class of non-linear hyperbolic systems, which requires that a time-stepping algorithm be introduced. Chavent and salzano[8] used a forward Euler method to march in time, but this had poor stability properties; it is stable for very restrictive CFL numbers. The restriction upon the CFL number would make it a restrictive condition for hyperbolic problems[2], because the CFL number is of the order of the the square-root of the grid size, which is very impractical.

The first steps towards developing a method which overcame the time-stepping problems were taken by Cockburn and Shu, who used a TVB Runge-Kutta devised by Shu and Osher[61]. The method subsequently became known as Runge-Kutta Discontinuous Galerkin (RKDG), which has since been applied extensively. Although the Runge-Kutta method has remained popular, alternative methods are being investigated. Qui *et.al*[55] developed a Lax-Wendroff type method for Discontinuous Galerkin which is more compact than the existing Runge-Kutta method. Unfortunately, the Lax-Wendroff method is more complicated to program than the popular TVD Runge-Kutta method for multidimensional cases.

In [10], Cockburn and Shu investigated a 1D single component scalar equation, which was extended to a 1D multicomponent scalar problem[9]. Continued work by Cockburn and Shu was focused on establishing a multidimensional RKDG theory[12], and an RKDG theory for convection-dominated problems[11]. Furthermore, articles have been written to give a wider audience introduction to the RKDG method[79]

The success of Discontinuous Galerkin can be illustrated by the areas to which it has been applied: Magnetohydrodynamics[14, 71], Plasma physics[42], Fluid mechanics[4,

36, 44] and Solid mechanics[66].

The greatest challenge faced in extending DG to higher dimensions is the *limiter*, which is required to suppress the oscillations which occur as a result of Godunov's theorem. Cockburn and Shu[13] stated that the development of a limiter in multidimensions is a difficult task. However, recent progress has been made in developing limiters, although these are applicable for 2D problems[70].

It has also been noted that the choice of basis function plays no part in the accuracy of the algorithm [76]. The choice of basis functions can be demarcated into orthogonal and non-orthogonal basis functions. Orthogonal basis functions result in a diagonal mass matrix, which is trivial to invert. If a non-orthogonal basis function is chosen, then a method for inverting the mass matrix is required.

A novel approach to defining a set of basis functions is to carry out transformations to the *collapsed coordinate space*. This method maps the reference hexahedron to the reference tetrahedron, and is also applicable to grids constructed of prism and pyramids; making it possible to define DG for a hybrid grid[45]. A DG method involving basis functions defined using a collapsed coordinate transformation have been applied to 3D problems[16]. With this DG approach, two transformations are required: the first transformation maps the physical tetrahedron to the reference tetrahedron and the second transformation maps the reference tetrahedron to the reference hexahedron. The benefit of a *collapsed Cartesian* coordinate system, is that 1D quadratures methods and wrapped tensors(basis functions) can be applied to multidimensional domains[33]

Researchers have also investigated the application of high-resolution methods to DG. Most notably is the application of Weighted Essentially Non-Oscillatory (WENO) limiters, which has been done for both 2D problems, for both structured and unstructured grids[54, 76], and for 3D problems[47].

1.3 Objectives of the thesis

Finite volume method (FVM) represents, arguably, the most ubiquitous discretisation approach in the engineering computational fluid dynamics community. However, Discontinuous Galerkin has, as mentioned previously, been applied to a variety of flows and is becoming equally widespread, due to a combination of the DG method's ability to model hyperbolic PDE flows on unstructured grids and a fixed direct side stencil approach, thus combining the beneficial properties of FVM in relation to the resolution of discontinuities and sharp gradients, and increasing the spatial accuracy using a FEM approach.

It is evident there is a similarity between the two methods, and this is substantiated by the fact that 1st order accurate DG is identical to the 1st order accurate FVM.

The methods obviously differ when the order of accuracy is greater than 1st order. Firstly, the solution reconstruction is different. In FVM the need to construct a larger stencil, based on the neighbours of neighbours is required. However, the DG method is

more localised requiring the same stencil as the 1st order method (only the direct side neighbours), but an increase in the number of degrees of freedom and the inclusion of a volume quadrature.

Although it is possible to obtain a discretisation of the same order of grid convergence with both methods, the properties of the asymptotic convergence range, and hence the actual order of accuracy demonstrated through the simulations differs. In this thesis we explore the similarities and differences between the Finite Volume and Discontinuous Galerkin solutions of the same order of grid convergence, within the context of the same CFD solver. The latter ensures that the manipulations of geometrical data are performed in the same way, the integrals are solved using the same quadrature points and the resolution of the discontinuity is obtained through the same Riemann solver - enabling a direct comparison of discretisation efficiency.

The majority of comparative studies of DG have been conducted in 1D and 2D and these have been done for a variety of problems, and to a variety of numerical methods. [60] compared FDM-WENO, FVM-WENO and DG-TVB in 2D for hyperbolic problems with strong shocks, the author showed that each method had particular strengths and weaknesses, in particular the lack of robustness of the (DG) TVB limiter when compared to WENO, and the ideal use of the DG method for problems with smooth solutions. [29] studied the comparison of high-order methods for oceanic applications in 2D, the DG method performed with relative efficiency when compared to the other numerical methods. Similarly [51] compared DG with FVM, but for a inviscid and viscous problems in 2D on a Cartesian grid upto 4th order. This comparison like the previous comparison, showed the effectiveness of using DG methods, especially for smooth solutions. [14] conducted a 3D study of DG, however, this was done for a structured hexahedral grid (not for tetrahedral cells we are considering in this thesis), and the study compared the DG method to the FDM (not to the FVM we are considering in this thesis) - it was shown like the previous study that the DG method performed better than the FDM. [74] conducted a 1D study on the DG method compared to the spectral finite volume method for the simple advection eqation, it was shown that the DG required a smaller CFL number for stability but produced an error which was also smaller. In [75] a 2D comparative study was conducted on an unstructured grids for problems which required limiting, for DG a TVB limiter was utilised whereas for FVM a WENO method was used - It was shown that the FVM-WENO approach produced a high error and was slower than the DG-TVB approach. From these comparative studies between DG and other numerical methods, the general conclusion states that the DG method performs well for smooth solutions, and when a TVB limiter is utilised performs well for discontinuous solutions too, although the size of the CFL number for DG is very restrictive.

To make the comparison unbiased as possible, a number of simplifications are applied. Firstly, an almost uniform smooth tetrahedral mesh is selected for the testcases in order to underline the observed differences in convergence. Secondly, the comparison is restricted to schemes of up to 3rd order of accuracy in order to evaluate the scenario corresponding to the majority of commercially available engineering FVM CFD

solvers. Finally, none of the aforementioned DG and FVM comparisons used an identical limiter for discontinuous solutions - in this thesis we will use a Barth-Jespersen limiter which is applied identically to both DG and FVM, unlike the TVB and WENO limiters, thus keeping the comparison as close as possible.

The objectives of this thesis can be summarised as following:

1) For a comparative study of FVM and DG in 3D the order of grid convergence is a vital indicator of the spatial accuracy of the method. This will be analysed for scalar testcases and a systems testcase (Euler equations) for problems which are sufficiently smooth (for example a sine wave on a fine mesh), thus not requiring a limiter function. From this study an convergence to the nominal order of accuracy is expected.

2) It has been observed that TVD limiters, which are routinely applied when there is chance of Gibb's oscillations forming, degrade the convergence of a numerical scheme, especially in the smooth region of a flow. An identical limiter is sought for both FVM and DG, to keep the comparison as close as possible, which is applied to a smooth problem to observe the degradation of the convergence.

3) In addition to the application of the TVD limiter to smooth problems, the limiter is applied to problems where the inclusion of a limiter to suppress oscillations, (which at times can cause the CFD solver to crash or produce non-physical results such as negative density) is necessary .

4) Another pertinent observation is the computational time of the numerical methods, this is an important component of any numerical method, especially in CFD where simulations can require a large amount of time to converge.

Both the methods are implemented on the same solver platform, thus giving the possibility to reuse shared features of the method, resulting in identical computational and discretisation round-off errors when possible .

1.3.1 Structure of the thesis

1) - The initial part of the thesis establishes the theory of the Discontinuous Galerkin method. The theory is firstly proposed in the 1D case, identifying the main areas of the method, this is then extended to multidimensions. The theory includes the choice of basis functions and modification required for them to be used on the FVM solver, the evaluation of the integrals, the initialisaiton of DG and the limiter required to suppress oscillations.

2) - The advection and Euler equations are solved for a constant initial condition. This is particularly important for verifying the derivatives of the basis functions and the volume quadratures(which are only present in high-order DG)

3) - The linear advection equation with initial conditions $sin(2\pi x)sin(2\pi y)sin(2\pi z)$, $sin(4\pi x)sin(4\pi y)sin(4\pi z)$, $sin(6\pi x)sin(6\pi y)sin(6\pi z)$ are solved on four grids of increasing grid points to establish a convergence analysis. The CFL number is also altered to observe the effect of stability and convergence of the FVM and DG method.

4) A manufactured method of solution is applied to the Euler equations for a sufficiently smooth density profile, which possess an analytical solution, therefore making it possible to conduct a nominal order of accuracy study.

5) Having verified the DG method using the advection equation and Euler equations, it is possible to investigate testcases where it is difficult to evaluate a nominal order of accuracy. These are testcases which involve a TVD limiter, which degrades the convergence of the scheme. In the first instance, to test the robustness of the DG solver, an implosion of a sphere of gas is investigated. The purpose of the implosion testcase is not to act as a comparison between the DG and FVM, as an analytical solution is very difficult to derive.

6) - The successful demonstration of the DG solver to handle the modelling of the shock collisions in the implosion testcase using a TVD limiter allows the DG solver to be applied to other testcase where a limiter is necessary. Sod's shocktube case is investigated on a rectangular domain, the flow exhibits all three wave solutions for an hyperbolic PDE. DG and FVM are applied to this testcase, this is the first comparative testcase involving a limiter which is applied to a shock problem, in this thesis. For the sake of completeness of the comparison, an identical limiter is utilised(the only such limiter on an unstructured grid) the Barth-Jespersen limiter.

7) - To investigate the comparison of FVM and DG with respect to an analytical solution in a 3D setting, an explosion testcase is chosen. A sphere of gas of high pressure at the centre of cubic domain surrounded by a gas of low pressure. This can be viewed as a 3D extension of the shocktube problem.

8) The results from the testcases (**2-7**) are discussed and a conclusion is presented at the end of the thesis.

1.4 Publication

Published

A.F. Antoniadis, K.H.Iqbal, E.Shapiro, N.Asproulis and D.Drikakis. *Comparison of High-Order Finite Volume and Discontinuous Galerkin Methods on 3D Unstructured Grids*, ICNAAM 2011

CHAPTER 2

Discontinuous Galerkin

2.1 1D Discontinuous Galerkin

The 1D Discontinuous Galerkin method for hyperbolic problems represents the main constituents of the method, and therefore is an ideal starting point before extending the discussion to the multidimensional case. In the 1D discussion we present the main components of the method: Evaluation of the integrals using a quadrature method, the basis functions and the associated orthogonality, and the Runge-Kutta method. Firstly, the hyperbolic PDE is cast into a conservation form (eq: 2.1.1).

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0 \tag{2.1.1}$$

Where the $F(U)$ is the flux of the conserved variables and U is the vector of conserved variables. In the case of the scalar problem and Euler equations[1] in the 1D these would be, respectively:

$$F(U) = C_x.U$$

$$F(\mathbf{U}) = \begin{pmatrix} \rho u \\ p + \rho u^2 \\ u(E + p) \end{pmatrix},$$

The Conservation law (eq: 2.1.1) is cast into the weak form by multiplying by some function $q(x)$, and integrating over the domain interval $\Omega = [x_{j-1/2}, x_{j+1/2}]$.

$$\int_\Omega \frac{\partial U}{\partial t} q(x) d\Omega + \int_\Omega \frac{\partial F(U)}{\partial x} q(x) d\Omega = 0 \tag{2.1.2}$$

The Galerkin formulation is based upon setting the basis function in the weak formulation (eq: 2.1.2) equal to the test function, where $\phi(x)$ is a basis function.

[1] Please consult the Appendix for a review of the Euler equations

$$\phi(x) = q(x)$$

Thus the weak form becomes

$$\int_{\Omega} \frac{\partial U}{\partial t}\phi(x)d\Omega + \int_{\Omega} \frac{\partial F(U)}{\partial x}\phi(x)d\Omega = 0 \tag{2.1.3}$$

The conserved variables U are replaced with the approximate solution U_h (eq: 2.1.4)

$$\boxed{U_h = \sum_{l=0}^{Np-1} \hat{u}_l(t)\phi_l(\xi(\mathbf{x}))} \tag{2.1.4}$$

From which we get the weak form in terms of the Degrees Of Freedom (DOF)

$$\int_{\Omega} \frac{\partial U_h}{\partial t}\phi(x)d\Omega + \int_{\Omega} \frac{\partial F(U_h)}{\partial x}\phi(x)d\Omega = 0 \tag{2.1.5}$$

Where $\hat{u}_l$ are the DOF - the l subscripts denote the order of the polynomial. The number of DOF per conserved variable in 1D is directly proportional to the order of the scheme i.e. a 1st order scheme has 1 DOF per conserved variable, a 2^{nd} scheme has 2 DOF per conserved variable , a 3^{rd} order has 3 DOF per conserved variable and so forth.

The choice of the basis functions is ambiguous, and has no effect on the overall results of the DG method[12]. Both Orthogonal and non-Orthogonal basis functions have been used. In this thesis we use the orthogonal Jacobi polynomials, $P_l^{\alpha,\beta}$. In 1D we use the Legendre polynomials which are a generalisation of the Jacobi polynomials (by setting the $\alpha = \beta = 0$), $P_l^{0,0}$, therefore, the basis function become the

$$\phi_l = P_l^{0,0}$$

The Legendre polynomials for upto 6th Order ($n \leq 5$) accuracy are presented in **table: 2.1** and **fig: 2.1**.

Table 2.1: Legendre polynomials for n≤ 5

Order (l)	$P_l^{0,0}$
0	1
1	x
2	$\frac{1}{2}(3x^2 - 1)$
3	$\frac{1}{2}(5x^3 - 3x)$
4	$\frac{1}{8}(35x^4 - 30x^2 + 3)$
5	$\frac{1}{8}(63x^5 - 70x^3 + 15x)$
⋮	⋮

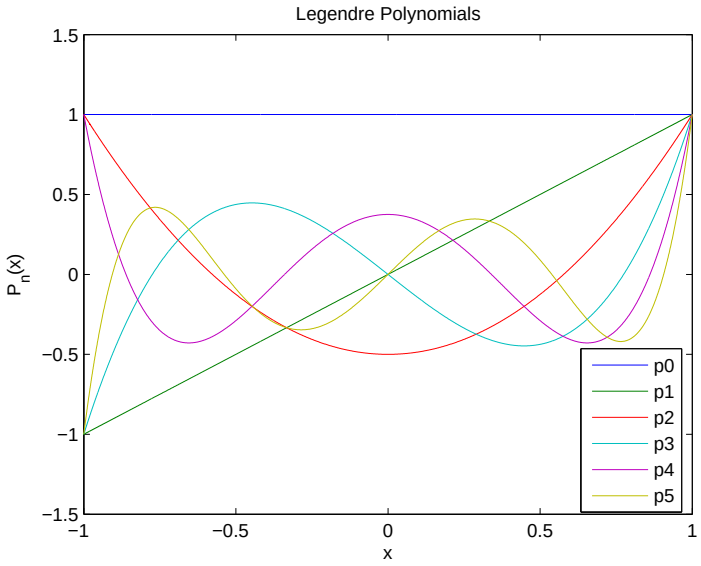


Figure 2.1: Legendre Polynomials for n≤5

The weak formulation in terms of DOF in its current form is not ideal for computation because of the partial derivative. To continue with the derivation of the 1D DG method, we apply the divergence theorem to the integral containing the conservative flux vector $F(U)$ (eq: 2.1.5), from which we retrieve the following PDE.

$$\int_\Omega \frac{\partial U_h}{\partial t}\phi(x)d\Omega - \int_\Omega \frac{d\phi(\xi(\mathbf{x}))}{dx}F(U_h)d\Omega + [\hat{F}(U_h)]_{i-1/2}^{i+1/2} = 0 \tag{2.1.6}$$

Now the integral containing the conservative flux vector has been separated into two parts. Firstly we have an integral containing the conservative flux vector

$$\int_\Omega \frac{d\phi(\xi(\mathbf{x}))}{dx}F(U_h)d\Omega$$

This is now in terms of an ordinary derivative and not a partial derivative and is sometimes referred to as the *volume integral* as it encompasses the entire volume of the element. The volume integral is solved by a Gaussian quadrature method, buy mapping the integral from the physical space x to the reference space ξ using the transformation

$$\xi = \frac{2(x - x_i)}{\Delta_i}$$

thus transforming the integral to the space [-1,1] it possible to use the Gaussian quadrature methods.

$$\int_\Omega \frac{d\phi(\xi(\mathbf{x}))}{dx}F(U_h)d\Omega = \frac{2}{\Delta_i}\int_{-1}^{1} \frac{d\phi(\xi)}{d\xi}F(U_h)d\hat{\Omega}$$

The second part is called the *numerical flux integral*, where the conservative flux F is replaced by a numerical flux $\hat{F}$

$$[\hat{F}(U_h)]_{i-1/2}^{i+1/2}$$

Although in the case of 1D DG it is merely at the end points of the line element ($\Omega = [x_{i-1/2}, x_{i+1/2}]$). The numerical flux is the only means by which elements can communicate with one another, due to the discontinuity between them. Such problems, where a discontinuity separates piecewise data is called a *Riemann problem*. For a simple advection problem we are considering in this chapter, a simple upwind Godunov flux will suffice. A comprehensive overview of which can be found in[68].

In the 2D and 3D cases a integral would have to be solved. The dimensions of the numerical flux integral is one less than the dimension of the space we are working in i.e in 3D space the numerical flux integral is a 2D integral.

To change the partial derivative to an ordinary derivative in the temporal integral we expand the approximate solution U_h, thus completing the transition from the partial differential form of the weak formulation to the semi-discrete form. Such that the Runga-Kutta method can be used to solve the semi-discrete form.

$$\int_\Omega \frac{\partial U_h}{\partial t}\phi(x)d\Omega = \int_\Omega \frac{\partial}{\partial t}\sum_{l=0}^{Np-1}\hat{u}_l(t)\phi_l(\xi(\mathbf{x}))\phi_m(\xi(\mathbf{x}))d\Omega$$

As the degree of freedom is only time-dependent it can be factored out of the integral

$$\frac{d}{dt}\hat{u}_l(t)\int_\Omega \phi_l(\xi(\mathbf{x}))\phi_m(\xi(\mathbf{x}))d\Omega$$

the Legendre polynomials are orthogonal in the space $[-1, 1]$, which will become our reference space. Therefore transforming this integral to the reference space $\hat{\Omega}$

$$\mathbf{J}.\frac{d}{dt}\hat{u}_l(t)\int_\Omega \phi_l(\xi)\phi_m(\xi)d\Omega$$

where $\mathbf{J}$ is the Jacobian of the transformation. the basis functions are orthogonal, and the orthogonality property applies, namely

$$\int_{\hat{\Omega}} \phi_l(\xi)\phi_m(\xi) = \delta_{lm}a_l$$

where δ_{lm} is the Kronecker delta and a_l is the norm

and the temporal integral becomes

$$\mathbf{J}.\mathbf{M}\frac{d}{dt}\hat{u}_l(t)$$

where **M** is the diagonal matrix of the norms of the basis functions (this is often called the mass matrix).

$$\mathbf{M} = \begin{pmatrix} a_0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & a_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_2 & 0 & 0 & 0 \\ \vdots & 0 & 0 & a_3 & 0 & \vdots \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 0 & a_{Np-1} \end{pmatrix}$$

the inversion of the diagonal mass matrix is a trivial process, and is simply

$$\mathbf{M}^{-1} = \begin{pmatrix} 1/a_0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1/a_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/a_2 & 0 & 0 & 0 \\ \vdots & 0 & 0 & 1/a_3 & 0 & \vdots \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1/a_{Np-1} \end{pmatrix}$$

The final form (semi-discrete)

is

$$\frac{d}{dt}\hat{u}_l(t) = \frac{1}{\mathbf{J}}\mathbf{M}^{-1}\left(\int_\Omega \frac{d\phi(\xi(\mathbf{x}))}{dx} F(U_h) - [F(U_h)]_{i-1/2}^{i+1/2}\right) \tag{2.1.7}$$

This semi discrete form is solved using a Runge-Kutta method[2].

2.2 1D Testcase

To demonstrate the 1D DG method, an advection equation is solved with initial condition

$$U_0 = sin(2\pi x)$$

This is solved for upto 6th order spatial accuracy (p_5) on 3 grids with elements ($N_G = 20, 40, 80$), on a spatial domain $[0, 1]$. Periodic boundary conditions are specified at the beginning and end of the domain (at 0 and 1), and the advection velocity is set to $C_x = 1$, therefore the time it takes for one period on the specified domain and advection velocity with the above IC is $T = 1$. The order of the Runge-Kutta method (the temporal order) is set equal to the order spatial accuracy. So for 1st order accuracy (P_0) we use a 1st order Runge-Kutta method (Euler's method), for 2nd order accuracy (P_1) a 2nd order method Runge-Kutta is used, and so forth. The size of the CFL number is dictated by[2]

$$CFL = \frac{1}{2l+1} \tag{2.2.1}$$

[2]Please consult the appendix for an overview of Runge-Kutta methods

The results of the convergence study is tabled in (**Table: 2.2**) which shows the nominal order of accuracy (L_∞ and L_1). L_0 is the maximum error of the error between the analytical solution (U_{Anl}) and the numerical solution U_h and L_1 is the defined as

$$L_1 = \frac{1}{N_E} \sum_{i=1}^{N_E} |U_h - U_{Anl}|$$

Table 2.2: 1D DG numerical convergence rates for linear advection at T=1 for IC $U_0 = sin(2\pi x)$. $CFL = \frac{1}{2l+1}$ and Runge-Kutta order equals spatial order

	1^{st} order			
N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	1.35E-1	1.06E-1	-	-
40	6.80E-2	5.30E-2	0.99	1.00
80	3.39E-2	2.630E-1	1.01	1.00
	2^{nd} order			
N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	1.47E-2	1.63E-2	-	-
40	4.01E-3	4.04E-3	1.87	2.01
80	1.05E-3	1.00E-3	1.93	2.00
	3^{rd} order			
N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	3.67E-4	1.78E-4	-	-
40	4.63E-5	2.20E-5	2.99	3.02
80	5.83E-6	2.73E-6	2.99	3.01
	4^{th} order			
N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	7.19E-6	3.27E-6	-	-
40	4.55E-7	2.04E-7	3.98	4.00
80	2.85E-8	1.28E-8	4.00	4.00
	5^{th} order			
N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	1.15E-7	4.73E-8	-	-
40	3.54E-9	1.46E-9	5.02	5.02
80	1.10E-10	4.56E-11	5.00	5.00
	6^{th} order			
N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	1.43E-9	6.02E-10	-	-
40	2.25E-11	9.41E-12	5.99	6.00
80	3.51E-13	1.82E-13	6.00	5.69

The results for the testcase are illustrated in **Figure 2.2, Figure 2.2**, which show the convergence to the nominal order of accuracy.

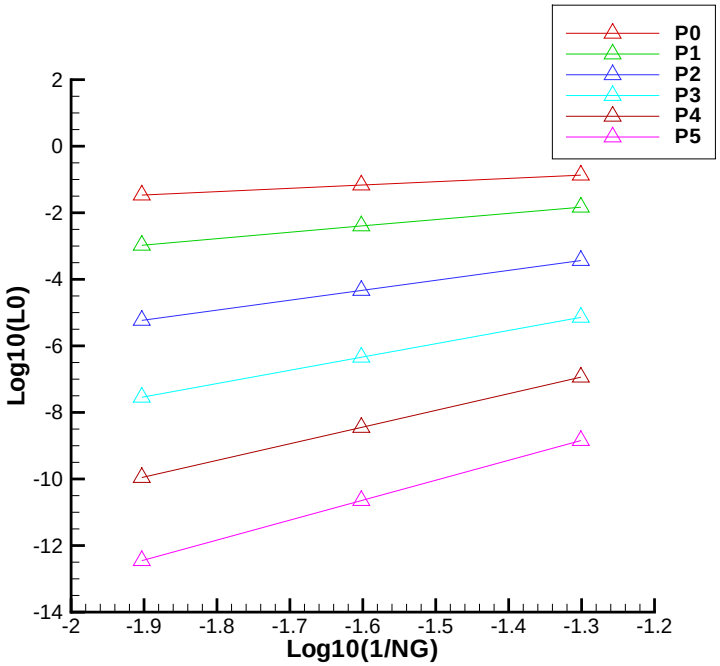


Figure 2.2: 1D DG L0 convergence for linear advection (upto 6^{th} order) at T=1 for IC $U_0 = sin(2\pi x)$

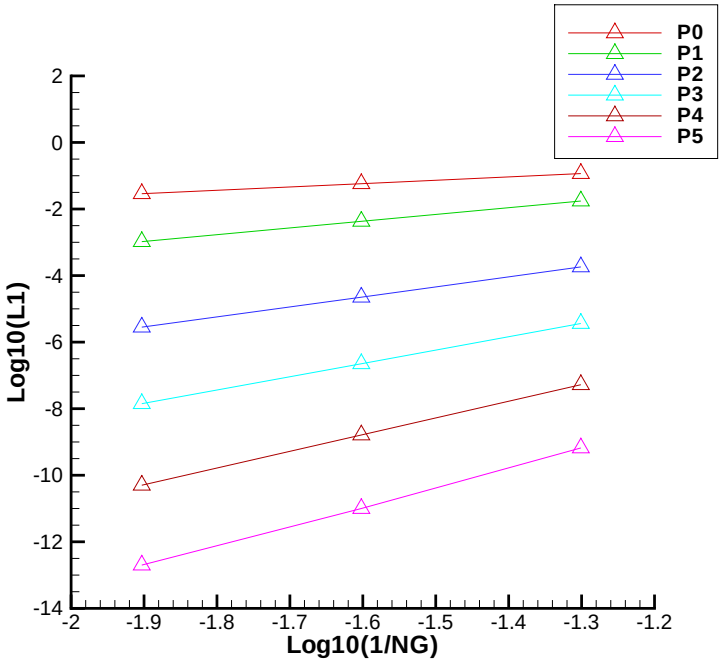


Figure 2.3: 1D DG L1 convergence for linear advection (upto 6^{th} order) at T=1 for IC $U_0 = sin(2\pi x)$

Given the very restrictive condition employed in the above problem: that the order of accuracy of the Runge-Kutta method equals the spatial order of accuracy, for example a 6th order Runge-Kutta was used for 6th order spatial accuracy (P_5), a natural question would be to ask if using a maximum of a 4th order Runge-Kutta would effect the convergence of method as witnessed in table: 2.2? Clearly in a 3D setting a 6th order Runge-Kutta would be infeasible due to the high-storage costs. Employing a maximum 4th order Runge-Kutta method in the following study, we investigate the convergence rates for upto 6th Order spatial accuracy (P_5),using the data provided in [13], which gives the CFL numbers for combination of the order of the RK and DG methods. The 1st order ($RK1$) and 2nd order ($RK2$) Runge-Kutta methods are only applicable to 1st and 2nd order spatial accuracy as stated in[13], and therefore will not be considered.

2nd Order spatial accuracy (P_1) The approximation for the maximum CFL number for $RK2$ is 0.333^* given by eq: 2.2 which is used for $RK2$ accuracy. Increasing the RK order to 3rd and 4th order we observed the following trends in the nominal order of accuracy, the maximum CFL which could be used with for P_1-$RK3$ was **0.409** **Table: 2.3**,and P_1-$RK4$ was **0.464** (**Table: 2.4**)

Table 2.3: P_1, $RK3$ for $CFL = 0.409$

N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	1.01E-2	8.41E-3	-	-
40	2.94E-3	1.94E-3	2.11	1.78
80	7.90E-4	4.67E-4	2.066	1.90

Table 2.4: P_1, $RK4$ for $CFL = 0.464$

N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	1.11E-2	8.49E-3	-	-
40	3.30E-3	2.13E-3	1.74	2.00
80	8.61E-4	5.10E-4	1.94	2.06

3nd Order spatial accuracy (P_2) The approximation for the maximum CFL number for $RK3$ is 0.2 given by eq: 2.2. Increasing the RK order to 4th Order we observed the following trends in the nominal order of accuracy, the maximum CFL which could be used with for P_2-$RK4$ was **0.235** table: 2.5

Table 2.5: P_2, $RK4$ for $CFL = 0.235$

N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	3.66E-4	1.57E-4	-	-
40	4.61E-5	1.96E-5	2.99	3.00
80	5.78E-6	2.45E-6	3.00	3.00

4th Order spatial accuracy (P_3) The approximation for the maximum CFL number for $RK4$ is 0.142 given by eq: 2.2. Decreasing the RK order to $RK = 3$ we observed

the following trends in the nominal order of accuracy, the maximum CFL which could be used with for P_3-$RK3$ was **0.130** table: 2.6

Table 2.6: P_3, $RK3$ for $CFL = 0.130$

N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	2.50E-4	2.51E-5	-	-
40	2.68E-5	3.14E-6	3.22	2.99
80	3.07E-6	3.94E-7	3.13	2.99

5th Order spatial accuracy (P_4) The approximation for the maximum CFL number for $RK5$ is 0.111^* given by eq: 2.2. Decreasing the RK order to 3rd and 4th order we observed the following trends in the nominal order of accuracy, the maximum CFL which could be used with for P_4-$RK3$ was **0.089** table: 2.7,and P_4-$RK4$ was **0.1** table: 2.8

Table 2.7: P_1, $RK3$ for $CFL = 0.089$

N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	5.73E-6	8.04E-6	-	-
40	7.15E-7	1.01E-6	3.00	3.00
80	8.94E-8	1.26E-7	3.006	3.00

Table 2.8: P_1, $RK4$ for $CFL = 0.1$

N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	1.65E-7	7.57E-8	-	-
40	6.73E-9	4.50E-9	4.62	4.07
80	3.10E-10	2.81E-10	4.44	4.00

6th Order spatial accuracy (P_5) The approximation for the maximum CFL number for $RK6$ is 0.0909^* given by eq: 2.2. Decreasing the RK order to 3rd and 4th order we observed the following trends in the nominal order of accuracy, the maximum CFL which could be used with for P_5-$RK3$ was **0.066** table: 2.9,and P_5-$RK4$ was **0.073** table: 2.10

Table 2.9: P_5, $RK3$ for $CFL = 0.066$

N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	2.33E-6	3.29E-6	-	-
40	2.92E-7	4.12E-7	3.00	3.00
80	3.65E-8	5.16E-8	3.006	3.00

Table 2.10: P_5, $RK4$ for $CFL = 0.073$

N_G	L_∞	L_1	O_{L_∞}	O_{L_1}
20	1.47E-8	2.04E-8	-	-
40	9.07E-10	1.28E-9	4.02	4.00
80	5.66E-11	8.00E-11	4.00	4.00

2.3 Summary of 1D method

The 1D Discontinuous Galerkin method was derived for a hyperbolic problem, which is demonstrated in the context of the the linear advection equation. Given the starting point of the derivation is a hyperbolic conservation law, the application of the 1D DG method to other hyperbolic PDEs will not require changing significant parts of the derivation.

The application of the 1D DG method to the linear advection equation for upto 6th order spatial accuracy (P_5) yielded the expected nominal order of convergence, when the order of accuracy of the Runge-Kutta method was equal to the spatial order of accuracy. However, if the order of accuracy of the Runge-Kutta method was *less than* the spatial order of accuracy then the nominal order of convergence is at most the order of the Runge-Kutta method. For example, the table: 2.7, 2.8 show the order of convergence when a 5th order method (P_4) uses Runge-Kutta methods of 4th and 3rd order respectively, then the nominal order of convergence is equal to the order of the Runge-Kutta method.

2.4 Nodal Basis functions

As was mentioned in [13], the choice of basis functions contributes little to the accuracy of the DG method. As a result both orthogonal[20, 45] and non-non-orthogonal[11–13] basis functions have been used. Apart from being orthogonal and non-orthogonal, basis functions can be modal and nodal. In this thesis we use the *modal* type basis function, with increasing order of accuracy, higher order polynomials are included in the approximation which are hierarchical in nature, this means that $\phi_0, \phi_1 \dots \phi_{N_p-2}$ are included along with the ϕ_{N_p-1} basis functions in $N_p - 1$ order approximation. This is illustrated in **Figure 2.1**. Unlike the FVM method, the DG method has more than one degree of freedom per variable for high-order accuracies. These degree of freedom are then reconstructed at the quadratures points for flux evaluation, and the centroid of the element for the final solution. The alternative to the modal basis functions are the *nodal* basis functions, which apart from increasing the mode of the polynomial, increase the nodes on which to evaluate the degrees of freedom. Nodal basis functions are not hierarchical meaning the basis functions for an $N_p - 1$ accuracy cannot be constructed from the $\phi_0, \phi_1 \dots \phi_{N_p-2}$ basis functions. Also, the reconstruction of the approximate solution is different. With the modal expansion, the position of the degrees of freedom $\hat{u}_l$ in the reference domain is not important, however in the nodal expansion this is important as a particular node corresponds to a particular basis function, hence information about the position of the degree of freedom is stored in the solution array.

$$
\begin{aligned}
U_h(x,t) &= \sum_{l=0}^{Np-1} \hat{u}_l(t)\phi_l(x) \qquad \text{Modal Expansion} \\
U_h(x,t) &= \sum_{l=0}^{Np-1} \hat{u}_l(x,t)\ell_l(x) \qquad \text{Nodal Expansion}
\end{aligned}
$$

To derive the basis functions for a general 1D domain we start with a representation of the approximate solution using a polynomial expansion

$$
U_h(\xi,t) = a_0 + a_1\xi_1 + a_2\xi_2 + a_3\xi_3 + ... \tag{2.4.1}
$$

Our aim is to establish the values of the unknown coefficient a_l. Assuming a reference domain of $[0, 1]$, although we could have worked in a physical domain or a reference domain of $[-1, 1]$. To illustrate how to construct a 2nd order set of coefficients we specify the approximate solution at end points 0 and 1 (any two points in the domain could have been chosen). At these points $\xi_{1,2} = 0, 1$ we get the following combinations

$$
\begin{aligned}
U_h(0,t) &= a_0 \Rightarrow a_0 = U_0 \\
U_h(1,t) &= a_0 + a_1 \Rightarrow a_1 = U_h(1,t) - U_h(1,t)
\end{aligned}
$$

Substituting the values of the coefficients a_l in the the polynomial expansion(equation

2.4.1) (for 2nd order expansion), and using the notation

$$\begin{aligned} U_h(0,t) &= \hat{u}_0 \\ U_h(1,t) &= \hat{u}_1 \end{aligned}$$

We obtain the following by substituting in equation 2.4.1

$$\begin{aligned} U_h(\xi,t) &= a_0 + a_1\xi_1 \\ U_h(\xi,t) &= \hat{u}_0 + (\hat{u}_1 - \hat{u}_0)\xi_1 \end{aligned}$$

rearranging there equations in terms of the degrees of freedom $\hat{u}_l$

$$\begin{aligned} U_h(\xi,t) &= \hat{u}_0(1-\xi_1) + \hat{u}_1\xi_1 \\ U_h(\xi,t) &= \hat{u}_0\ell_0 + \hat{u}_1\ell_1 \end{aligned}$$

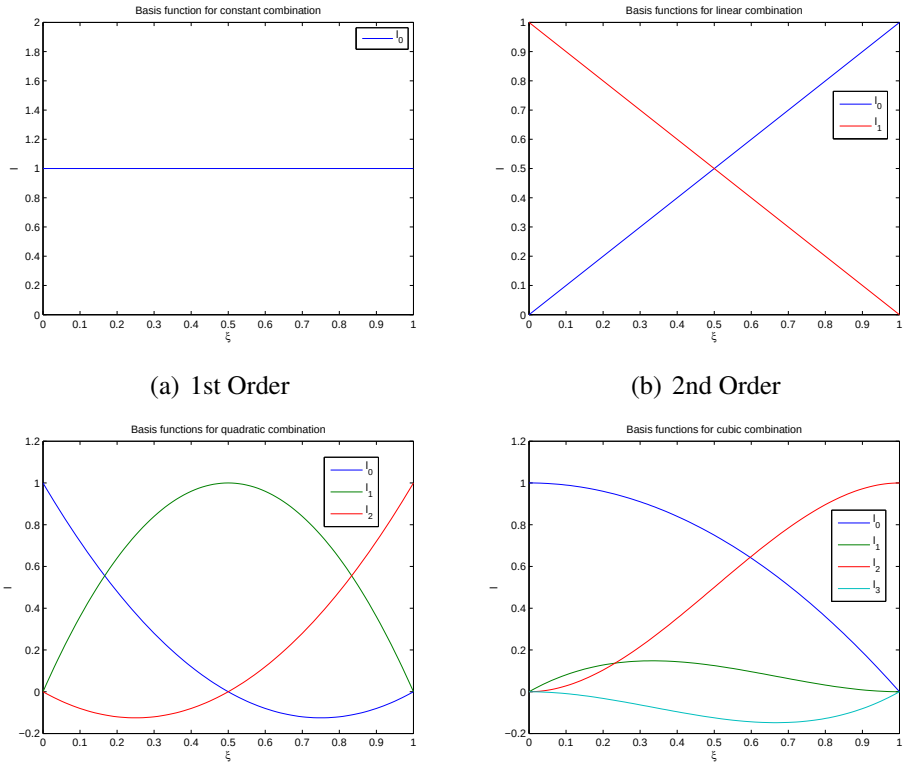
where the nodal basis functions in 2nd order (linear) approximation are given by

$$\begin{aligned} \ell_0 &= \xi - 1 \\ \ell_1 &= \xi \end{aligned}$$

Nodal basis functions can be derived for *nth* order accuracy - the number of basis functions (hence the number of nodes on the element) is proportional to the order of accuracy, therefore, 3rd order accuracy has three nodes and thus three basis functions. **Figure 2.4**, gives the illustration of the basis functions upto 4th order accuracy in the $[0, 1]$ domain, unlike the modal basis functions the nodal basis functions differ for each order of accuracy and we cannot use the 2nd order basis functions in a 3rd order approximate, as a result new 3rd order basis functions have to derived. An alternative to defining the basis functions using the above method is to use the Lagrangian polynomials

$$\ell_l(\xi) = \prod_{\substack{j=0 \\ l \neq j}} \frac{x - \xi_j}{\xi_i - \xi_j}$$

An in depth study of Nodal discontinuous Galerkin methods is given by[28].



(a) 1st Order

(b) 2nd Order

(c) 3rd Order

(d) 4th Order

Figure 2.4: 1D nodal basis functions defined in the reference space [0,1] for upto 4th order

2.5 3D Discontinuous Galerkin

A multidimensional hyperbolic conservation law is defined as

$$\frac{\partial U}{\partial t} + \nabla.\mathbf{F}(U) = 0$$

equally, by expanding the divergence operator, the hyperbolic conservation can be written as

$$\frac{\partial U}{\partial t} + \frac{\partial}{\partial x}F(U) + \frac{\partial}{\partial y}G(U) + \frac{\partial}{\partial z}H(U) = 0$$

where $\mathbf{F}$ is the flux vector defined as $(F(U), G(U), H(U))$, and U is the conserved variables.

For the scalar advection case the conserved variable is

$$U = (u)^T$$

with the fluxes $\mathbf{F}$ for the scalar advection being

$$F = C_x.u \quad G = C_y.u \quad H = C_z.u$$

C_x, C_y, C_z are the advection velocities in the x, y, z directions respectively.

For the three dimensional Euler equation the conserved variables are[3]

$$U = (\rho, \rho u, \rho v, \rho w, E)^T$$

As there are six unknowns and fives equations: pressure, density, the three velocity components and Energy. Therefore, the Euler equations are closed using an equation of state.

2.6 Weak Formulation

The starting point of the DG formulation is the derivation of the weak formulation, this is achieved by multiplying the conservation law by a test function $q(x, y, z)$ and integrating over an element Ω

$$\int_\Omega \frac{\partial U}{\partial t} q(x, y, z) d\Omega + \int_\Omega \left(\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z}\right) q(x, y, z) d\Omega = 0 \tag{2.6.1}$$

the second integral in the above formulation is integrated by parts using the Green-Gauss theorem, resulting in the following form.

$$\int_\Omega \frac{\partial U}{\partial t} q(x, y, z) d\Omega - \int_\Omega (F, G, H).\nabla q(x, y, z) d\Omega + \int_{\partial\Omega} (F, G, H).\hat{\mathbf{n}}.q(x, y, z) d\partial\Omega = 0 \tag{2.6.2}$$

[3]The fluxes for the Euler equations have been included in the appendix.

For the galerkin formulation the test function q(x,y,z) is replaced with the basis function

$$q(x, y, z) = \phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))$$

and replacing the conserved variable U with the approximate solution U_h

$$\boxed{U_h = \sum_{l=0}^{Np-1} \hat{u}_l(t)\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))} \tag{2.6.3}$$

the terms $(F, G, H).\hat{\mathbf{n}}$ in the surface integral are replaced with the numerical fluxes $\hat{F}, \hat{G}, \hat{H}$, where $\hat{\mathbf{F}}_h = (\hat{F}_h, \hat{G}_h, \hat{H}_h) = (\hat{F}(U_h), \hat{G}(U_h), \hat{H}(U_h))$. The choice for the numerical flux will be discussed in the proceeding sections.

$$\begin{aligned}\int_\Omega \frac{\partial U_h}{\partial t}\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))d\Omega \\ - \int_\Omega (F_h, G_h, H_h).\nabla\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))d\Omega \\ + \int_{\partial\Omega} (\hat{F}_h, \hat{G}_h, \hat{H}_h)\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))dS = 0\end{aligned} \tag{2.6.4}$$

with $\mathbf{F}_h = (F_h, G_h, H_h) = (F(U_h), G(U_h), H(U_h))$.

Equation (2.6.4) is referred to as the **weak Galerkin formulation**. This equation is not useful in its current form and has to be reduced to an ODE (Ordinary Differential Equation) to be more computational friendly. Firstly, the weak formulation is altered to reflect the type of element that is being considered. As we are considering only tetrahedrons, we change the surface integral to sum over all four faces. For other elements the surface integral has to changed respectively; six faces for hexahedrons, and fives faces for prism and pyramids. Another complicating factor is the shape of the face, in the tetrahedron all four faces are triangles. Whilst an hexahedron is constructed of 6 rectangles, and prisms and pyramids are a mixture of both rectangles and triangles.

$$\begin{aligned}\int_\Omega \frac{\partial U_h}{\partial t}\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))d\Omega \\ - \int_\Omega (F_h, G_h, H_h).\nabla\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))d\Omega \\ + \sum_{e\in\partial\Omega}^{4} \int_e (\hat{F}_h, \hat{G}_h, \hat{H}_h)\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))dS = 0\end{aligned} \tag{2.6.5}$$

To complete the derivation of the ODE from the weak form (2.6.5) we exploit the approximate solution U_h in the first integral.

$$\begin{aligned}
&\int_\Omega \frac{\partial U_h}{\partial t}\phi_l(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))d\Omega \\
&\quad= \int_\Omega \frac{\partial}{\partial t}\Big(\sum_{l=0}^{Np-1}\hat{u}_l(t)\phi_l(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))\Big)\phi_m(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))d\Omega \\
&\quad= \frac{d\hat{u}_l(t)}{dt}\int_\Omega \phi_l(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))\phi_m(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))d\Omega
\end{aligned}$$

the weak Galerkin form has now been reduced to an ODE, and can be solved using a time-stepping algorithm.

$$\begin{aligned}
&\frac{d\hat{u}_l(t)}{dt}\underbrace{\int_\Omega \phi_l(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))\phi_m(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))d\Omega}_{\text{Mass matrix}} \\
&\quad\underbrace{-\int_\Omega (F_h,G_h,H_h).\nabla\phi_l(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))d\Omega}_{\text{Volume integral}} \\
&\quad\underbrace{+\sum_{e\in\partial\Omega}^{4}\int_e (\hat{F}_h,\hat{G}_h,\hat{H}_h)\phi_l(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))dS}_{\text{Surface integral}} = 0
\end{aligned} \tag{2.6.6}$$

Equation(2.6.6) can be written in the concise form.

$$\boxed{\mathbf{M}\frac{d\hat{u}_l(t)}{dt} = \mathbf{L_h}(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))}$$

Where $\mathbf{L_h}$ is the right hand consisting of the volume and surface integral.

$$\boxed{\mathbf{L_h} = \int_\Omega (F_h,G_h,H_h).\nabla\phi_l(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))d\Omega - \sum_{e\in\partial\Omega}^{4}\int_e (\hat{F}_h,\hat{G}_h,\hat{H}_h)\phi_l(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))dS}$$

and $\mathbf{M}$ is the diagonal mass matrix.

$$\boxed{\mathbf{M} = \int_\Omega \phi_l(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))\phi_m(\xi(\mathbf{x}),\eta(\mathbf{x}),\zeta(\mathbf{x}))d\Omega}$$

Although the ODE form of the DG formulation has been established, there still remains unanswered questions. The main ones being:

1. The choice of the basis functions
2. The choice of the numerical flux
3. The method for solving the ODE
4. The method for evaluating the integrals

These points will be addressed in the proceeding sections.

2.7 Basis functions

In the study of continuous and discontinuous finite element methods there are two types of basis functions that can be utilised - non-hierarchical and hierarchical. Let us consider the non-hierarchical basis functions, which have to obey the following two properties[41].

1) The basis function ϕ_l is defined at a node i on element x_j and 0 everywhere else on the element

2) The basis function can only take non-zero values on the element being considered, and is zero on the elements not being considered.

$$\phi(x_j) = \begin{cases} \neq 0 & \text{if } x_j \in \Omega_j \\ = 0 & \text{if } x_j \notin \Omega_j \end{cases}$$

There is an extensive list of non-hierarchical basis functions for various elements in multidimensional [78], and these have been widely used in the continuous finite element method and also in three-dimensional discontinuous Galerkin.

The other category of basis function as mentioned above is of the hierarchical type. These basis functions only obey the second property i.e the basis function is only non-zero on the element being considered. The hierarchical basis functions are so named because the basis functions used in the P_{k+1} scheme contain the basis functions used in the P_k scheme. For example, a 3D tetrahedral requires 4 degrees of freedom and hence 4 basis functions for the P_1 and, 10 degrees of freedom and basis functions for the P_2 case. The basis functions used in the P_1 case form the first 4 basis functions of the P_2 in mathematical notation this can be represented as

$$\{\phi_l\}_0 \subset \{\phi_l\}_1 \subset \cdots \subset \{\phi_l\}_{K+1}$$

This can also be viewed as the set of P_0 basis functions are contained with the set of P_1 basis functions; the set of P_1 basis are contained with the set of P_2 basis and so forth.

Another feature of the basis functions to address is the orthogonality. Both hierarchical and non-hierarchical basis functions can be constructed to become a set of orthogonal basis functions, this is achieved by using the Gram-Schmidt method. The orthogonality of the basis functions will result in a diagonal mass matrix $\mathbf{M}$, which is trivial to invert.

The basis functions can be derived in either the physical space Ω [12, 76] or the reference space $\hat{\Omega}$ [20, 33, 73]. As the reference element is fixed, it easier to derive the basis functions in the reference space. To construction the basis functions in the reference space we simply map the element in the physical space to the reference triangle, and carry out all operations in the reference space. In the proceeding chapter we will show how this is achieved.

2.7.1 Basis function in $\hat{\Omega}$-space

In the $\hat{\Omega}$-space, where the physical tetrahedron has been transformed into the the standard tetrahedron in the [0,1] space, and the integration of the inner product is performed in this space.

To derive the orthogonal hierarchical basis function firstly consider the set B of monomial polynomials

$$B = \{b_i | i = 1, 2..., Np\} = \{\xi, \eta, \zeta, \xi\eta, \xi\zeta, ...\}$$

From which we want to construct a set of orthogonal basis functions G,

$$G = \{g_i | i = 1, 2..., Np\}$$

by applying the Gram-Schmidt method to the set of monomials B, we get

$$g_i = \frac{b_i - \sum_{k=1}^{i-1} \langle b_i, g_k \rangle g_k}{\sqrt{(b_i, b_i - \sum_{k=1}^{i-1} \langle b_i, g_k \rangle g_k}}$$

If the upper limit is less than the lower limit then the summation is taken to be zero. The monomials can be written in the general form

$$b_k = \xi^{\alpha(k)} \eta^{\beta(k)} \zeta^{\delta(k)}$$

therefore the inner product becomes

$$\langle b_i, b_k \rangle = \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} (\xi^{\alpha(i)} \eta^{\beta(i)} \zeta^{\delta(i)}).(\xi^{\alpha(k)} \eta^{\beta(k)} \zeta^{\delta(k)}) d\xi d\eta d\zeta$$

which can be written simply (by collecting the ξ, η, ζ terms)

$$\langle b_i, b_k \rangle = \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} (\xi^m \eta^n \zeta^l) d\xi d\eta d\zeta \tag{2.7.1}$$

where

$$m = \alpha(i) + \alpha(k); \quad n = \beta(i) + \beta(k); \quad l = \delta(i) + \delta(k)$$

the general solution for the inner product (2.7.1) can be established by repetitively using the binomial theorem, and the resulting answer is a set of orthogonal basis functions in terms of the Jacobi polynomials[28].

The Basis functions for the fours euclidean solids, which generally form the construction of an unstructured mesh[59].

Tetrahedral expansion:

$$\phi_{pqr}(\xi_1, \xi_2, \xi_3) = \tilde{\psi}_p^a(\eta_1) \tilde{\psi}_{pq}^b(\eta_2) \tilde{\psi}_{pqr}^c(\eta_3)$$

where

$$\eta_1 = \frac{2\xi_1}{1-\xi_2-\xi_3}, \quad \bar{\eta}_1 = \frac{2\xi_1}{1-\xi_3} - 1, \quad \eta_2 = \frac{2\xi_2}{1-\xi_3} - 1, \quad \eta_3 = -1 + 2\xi_3$$

now the tetrahedron basis function is defined in terms of the collapsed coordinates η, which are functions of the local space coordinates ξ. The basis functions above are defined in terms of Jacobi polynomials $P_p^{\alpha,\beta}(z)$

$$\tilde{\psi}_p^a(z) = P_p^{0,0}(z), \qquad \tilde{\psi}_{pq}^b(z) = \left(\frac{1-z}{2}\right)^p P_q^{2p+1,0}(z)$$

$$\tilde{\psi}_{pqr}^c(z) = \left(\frac{1-z}{2}\right)^{p+q} P_r^{2p+2q+2,0}(z)$$

when the parameters $\alpha = \beta = 0$ then the Jacobi polynomials $P_p^{\alpha,\beta}$ become the Legendre polynomials

$$P_p^{\alpha,\beta}(z) = P_p^{0,0}(z) = L_p(z)$$

and the use of the Legendre polynomials orthogonality property is made

$$\int_{-1}^{1} P_l(z)P_m(z)dz = \frac{2}{2l+1}\delta_{lm}$$

the orthogonality property of the Jacobi polynomials is

$$\int_{-1}^{1} (1-z)^\alpha (1+z)^\beta P_l^{\alpha,\beta}(z)P_m^{\alpha,\beta}(z)dz = \frac{2^{\alpha+\beta+1}}{2l+\alpha+\beta+1}\frac{\Gamma(l+\alpha+\beta)\Gamma(l+\alpha+\beta+1)}{n!\Gamma(n+\alpha+\beta+1)}\delta_{lm}$$

In the basis functions mentioned above the β parameter of the Jacobi polynomials is zero, and therefore the orthogonality property reduces to

$$\int_{-1}^{1} (1-z)^\alpha P_l^{\alpha,0}(z)P_m^{\alpha,0}(z)dz = \frac{2^{\alpha+1}}{2l+\alpha+1}\delta_{lm} \tag{2.7.2}$$

Where $(1-z)^\alpha$ is the weighting function, if we set $\alpha = 0$ we retrieve the orthogonality property for the Legendre polynomals used in the 1D setting.

NOTE

The basis functions given in journals such as[20, 59] and the book[33] collapse the coordinates of the reference hexahedron in a manner in which the reference pyramid, prism and tetrahedron are defined in the $[-1, 1]$ space. However, the above collapsed coordinates $\eta_1, \bar{\eta}_1, \eta_2, \eta_3$ have been modifed such that the reference hexahedron is defined in the $[0, 1]$ space, and as a result the collapsing of the coordinates will lead to the reference pyramid, prism and tetrahedron being defined in the $[0, 1]$ space. As a consequence of altering the reference space, the quadrature integration of the integrals is also effected. The quadrature method mentioned in [33] is reliant upon the radau and Labatto points which are defined in the $[-1, 1]$ space, these can be evaluated to a n^{th} order of accuracy using the software *R*, with the package *rJacobi*. For the evaluation of the basis functions in the $[0, 1]$ space the simplex points are used - this will be discussed in the proceeding section.

2.8 Initialisation of RKDG method

Let the initial condition be defined in the physical space Ω as

$$U(x, y, z, 0) = U_0(x, y, z)$$

replacing the left-hand side with the approximate solution

$$U_h(x, y, z, 0) = U_0(x, y, z) \tag{2.8.1}$$

where, as previously defined

$$U_h = \sum_{l=0}^{Np-1} \hat{u}_l(t)\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x})) \tag{2.8.2}$$

Casting this into the weak form, by multiplying equation (2.8.1) by the test function and integrating over the element of the domain.

$$\int_\Omega U_h(x, y, z, 0)\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))d\Omega = \int_\Omega U_0(x, y, z)\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))d\Omega \tag{2.8.3}$$

substituting (2.8.2) into (2.8.3) i.e the definition of U_h (the approximate solution)

$$\int_\Omega \Big(\sum_{l=0}^{Np-1} \hat{u}_l(0)\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x})) \Big)\phi_m(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))d\Omega$$
$$= \int_\Omega U_0(x, y, z)\phi_l(\xi_1(\mathbf{x}), \xi_2(\mathbf{x}), \xi_3(\mathbf{x}))d\hat{\Omega}$$

factoring out the degree of freedom,

$$\hat{u}_l(0).\int_\Omega \phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x})).\phi_m(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))d\Omega \tag{2.8.4}$$
$$= \int_\Omega U_0(x, y, z)\phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))d\hat{\Omega}$$

where the integral on the left hand side of (2.8.4) is the mass matrix

$$\mathbf{M} = \int_\Omega \phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x})).\phi_m(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))d\Omega$$

This integral is evaluated in the reference space $\hat{\Omega}$ (the space in which the basis functions are defined).

$$\mathbf{M} = \mathbf{J}.\int_{\hat{\Omega}} \phi_l(\xi, \eta, \zeta)\phi_m(\xi, \eta, \zeta)d\hat{\Omega} \tag{2.8.5}$$

where **J** is the Jacobian of the transformation. As the Jacobian is constant the orthogonality of the mass matrix will be preserved in the reference space[21]. The Basis functions, as we have already mentioned, are orthogonal in the reference space. Therefore the mass matrix (2.8.5) can be written as

$$\mathbf{M} = \mathbf{J}.a_l$$

where a_l is the norms[4].

Similarly the right hand of side is evaluated in the reference space, which becomes

$$\mathbf{J}.\int_{\hat{\Omega}} U_0(x(\xi,\eta,\zeta), y(\xi,\eta,\zeta), z(\xi,\eta,\zeta)).\phi_l(\xi,\eta,\zeta)d\hat{\Omega}$$

Finally the initialisation becomes

$$\hat{u}_l(0) = \frac{1}{a_l}\int_{\hat{\Omega}} U_0(x(\xi,\eta,\zeta), y(\xi,\eta,\zeta), z(\xi,\eta,\zeta)).\phi_l(\xi,\eta,\zeta)d\hat{\Omega}$$

2.9 Evaluation of the Integrals

In the previous section it was shown that in the semi-discrete derivation of the discontinuous Galerkin method, the need to evaluate integrals arise. These are most often carried out using quadrature methods, this process is called *numerical integration*. Numerical integration makes it possible to evaluate complicated integrals, which may not posses an analytical answer. Performing integration on elements with orthogonal edges: a straight line in 1-D, a square in 2-D and a cube in 3-D is a trivial it is merely the tensor product of the Gauss-Legendre quadrature method.

However, it is more complicated for simplexes such as triangles and tetrahedrons. Hammer et.al [25] made the first attempt to perform numerical integration on a simplex, and these were extended by Dunavant[19] and Zienkiewicz[77] for triangles and by Jinyun[31] and Keast[34] for tetrahedrons. Although these methods provide a useful way of evaluating the integral, they are limited. The highest order of accuracy available for numerical integration on a tetrahedron is 8[34]. As the order of accuracy of the DG method is effected by the order of the quadrature method it is envisaged that the aforementioned choice of quadrature points may limit the order to which we can extend the DG method.

To circumvent this problem the numerical integration proposed by Karniadakis and Sherwin[33] coupled with the quadrature points provided by the *R* numerical software (with the package *rJacobi*) give us the method of possible derivation of a DG method for an arbitrary high order of accuracy.

[4] the norms of the basis functions are listed in the appendix

Table 2.11: Gaussian quadrature points and weights for triangle in $[0,1]^2$ space from [19]

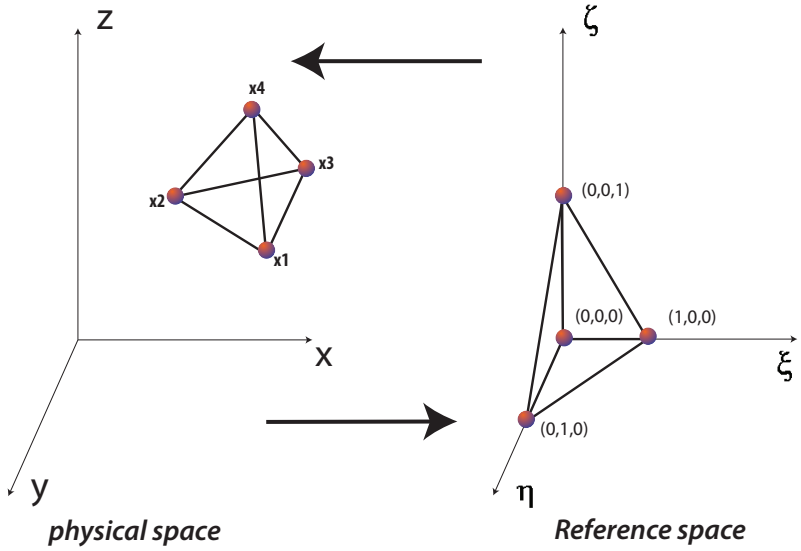
Order	ξ	η	ω
1	0.333333333333333	0.333333333333333	1.000000000000000
2	0.666666666666667	0.166666666666667	0.333333333333333
	0.166666666666667	0.166666666666667	0.333333333333333
	0.166666666666667	0.666666666666667	0.336666663333333
3	0.333333333333333	0.333333333333333	-0.56250000000000
	0.600000000000000	0.200000000000000	0.52083333333333
	0.200000000000000	0.600000000000000	0.52083333333333
	0.200000000000000	0.200000000000000	0.52083333333333
4	0.816847572980459	0.091576213509771	0.109951743655322
	0.091576213509771	0.816847572980459	0.109951743655322
	0.091576213509771	0.091576213509771	0.109951743655322
	0.108103018168070	0.445948490915965	0.223381589678011
	0.445948490915965	0.108103018168070	0.223381589678011
	0.445948490915965	0.445948490915965	0.223381589678011
5	0.333333333333333	0.333333333333333	0.225000000000000
	0.797426985353087	0.101386507323456	0.125939180544827
	0.101386507323456	0.797426985353087	0.125939180544827
	0.101386507323456	0.101386507323456	0.125939180544827
	0.059715871789769	0.470142064105115	0.132394152788506
	0.470142064105115	0.059715871789769	0.132394152788506
	0.470142064105115	0.470142064105115	0.132394152788506
6	0.501426509658179	0.249286745170910	0.116786275726379
	0.249286745170910	0.501426509658179	0.116786275726379
	0.249286745170910	0.249286745170910	0.116786275726379
	0.873821971016996	0.063089014491502	0.050844906370207
	0.063089014491502	0.873821971016996	0.050844906370207
	0.063089014491502	0.063089014491502	0.050844906370207
	0.310352451033784	0.053145049844817	0.082851075618374
	0.310352451033784	0.636502499121399	0.082851075618374
	0.053145049844817	0.636502499121399	0.082851075618374
	0.053145049844817	0.310352451033784	0.082851075618374
	0.636502499121399	0.310352451033784	0.082851075618374
	0.636502499121399	0.053145049844817	0.082851075618374

In this thesis we limit the order of accuracy to 2nd order, therefore, the simplex points of [19, 34] will suffice. The quadrature points are defined in the reference space, and points are mapped to the physical space, this is illustrated by figure (2.5)

Table 2.12: Gaussian quadrature points and weights for Tetrahedron in $[0, 1]^3$ space from [34]

Order	ξ	η	ζ	ω
1	0.25000000000000	0.25000000000000	0.25000000000000	1.00000000000000
2	0.25000000000000	0.25000000000000	0.25000000000000	-0.8000000000000
	0.50000000000000	0.16666666666667	0.16666666666667	0.4500000000000
	0.16666666666667	0.16666666666667	0.16666666666667	0.4500000000000
	0.16666666666667	0.16666666666667	0.50000000000000	0.4500000000000
	0.16666666666667	0.50000000000000	0.16666666666667	0.4500000000000
3	0.56843058419684	0.14385647193439	0.14385647193439	0.21776506988041
	0.14385647193439	0.14385647193439	0.14385647193439	0.21776506988041
	0.14385647193439	0.14385647193439	0.56843058419684	0.21776506988041
	0.14385647193439	0.56843058419684	0.14385647193439	0.21776506988041
	0.00000000000000	0.50000000000000	0.50000000000000	0.02148995341306
	0.50000000000000	0.00000000000000	0.50000000000000	0.02148995341306
	0.50000000000000	0.50000000000000	0.00000000000000	0.02148995341306
	0.50000000000000	0.00000000000000	0.00000000000000	0.02148995341306
	0.00000000000000	0.50000000000000	0.00000000000000	0.02148995341306
	0.00000000000000	0.00000000000000	0.50000000000000	0.02148995341306
4	0.25000000000000	0.25000000000000	0.25000000000000	-0.07893333333331
	0.78571428571429	0.07142857142857	0.07142857142857	0.045733333333333
	0.07142857142857	0.07142857142857	0.07142857142857	0.045733333333333
	0.07142857142857	0.07142857142857	0.78571428571429	0.045733333333333
	0.07142857142857	0.78571428571429	0.07142857142857	0.045733333333333
	0.10059642383320	0.39940357616680	0.39940357616680	0.149333333333333
	0.39940357616680	0.10059642383320	0.39940357616680	0.149333333333333
	0.39940357616680	0.39940357616680	0.10059642383320	0.149333333333333
	0.39940357616680	0.10059642383320	0.10059642383320	0.149333333333333
	0.10059642383320	0.39940357616680	0.10059642383320	0.149333333333333
	0.10059642383320	0.10059642383320	0.39940357616680	0.149333333333333
5	0.25000000000000	0.25000000000000	0.25000000000000	0.18170206858254
	0.00000000000000	0.33333333333333	0.33333333333333	0.03616071428571
	0.33333333333333	0.33333333333333	0.33333333333333	0.03616071428571
	0.33333333333333	0.33333333333333	0.00000000000000	0.03616071428571
	0.33333333333333	0.00000000000000	0.33333333333333	0.03616071428571
	0.72727272727272	0.09090909090909	0.09090909090909	0.06987149451617
	0.09090909090909	0.09090909090909	0.09090909090909	0.06987149451617
	0.09090909090909	0.09090909090909	0.72727272727272	0.06987149451617
	0.09090909090909	0.72727272727272	0.09090909090909	0.06987149451617
	0.43344984642634	0.06655015357366	0.06655015357366	0.06569484936832
	0.06655015357366	0.43344984642634	0.06655015357366	0.06569484936832
	0.06655015357366	0.06655015357366	0.43344984642634	0.06569484936832
	0.06655015357366	0.43344984642634	0.43344984642634	0.06569484936832
	0.43344984642634	0.06655015357366	0.43344984642634	0.06569484936832
	0.43344984642634	0.43344984642634	0.06655015357366	0.06569484936832

Figure 2.5: Transformation between the physical and reference space



Surface Integral

The surface integral, S_I, is defined as

$$S_I = \int_{\partial\Omega} \hat{F}(U_L^h, U_R^h, n).\phi_l(\xi_p^V(x,y,z), \eta_p^V(x,y,z), \zeta_p^V(x,y,z))dS$$

and the transformation of some point ξ_p^S, η_p^S from the reference space to the physical space for a triangle is given by:

$$\begin{aligned} x_p^S(\xi_p^S, \eta_p^S) &= x_1 + (x_2 - x_1)\xi_p^S + (x_3 - x_1)\eta_p^S \\ y_p^S(\xi_p^S, \eta_p^S) &= y_1 + (y_2 - y_1)\xi_p^S + (y_3 - y_1)\eta_p^S \\ z_p^S(\xi_p^S, \eta_p^S) &= z_1 + (z_2 - z_1)\xi_p^S + (z_3 - z_1)\eta_p^S \end{aligned} \tag{2.9.1}$$

To evaluate the surface integrals we map the quadrature points ξ_p^S, η_p^S from the reference triangle to the physical triangle, thus, the surface integral therefore becomes

$$S_I = \mathbf{J}_S . \int_{\partial\hat{\Omega}} \hat{F}(U_L^h, U_R^h, n)\phi_l(\xi_p^V(x_p^S, y_p^S, z_p^S), \eta_p^V(x_p^S, y_p^S, z_p^S), \zeta_p^V(x_p^S, y_p^S, z_p^S))d\hat{S}$$

where $x_p^S(\xi_p^S, \eta_p^S), y_p^S(\xi_p^S, \eta_p^S), z_p^S(\xi_p^S, \eta_p^S)$ are calculated using(2.9.1), and $\mathbf{J}_S$ is the Jacobian of the transformation for a triangle.

The quadrature approximation is given by the following expression, where $|S|$ is the surface area of the physical triangle and w_j is weight associated with the quadrature points. The surface area $|S|$ arises as a consequence of the type of the quadrature method. In the methods mentioned previously [19, 34], the quadrature approximation has to be multiplied by the area of the reference element. In this case the reference triangle is $\frac{1}{2}$, this results in the well-known expression $|S| = \frac{1}{2}\mathbf{J}_S$: the area of the triangular surface in the physical space.

$$S_I \approx |S| \sum_{j=1}^{J} \hat{F}\phi_l(\xi_p^S(x_{jp}^S, y_{jp}^S, z_{jp}^S), \eta_p^V(x_{jp}^S, y_{jp}^S, z_{jp}^S), \zeta_p^V(x_{jp}^S, y_{jp}^S, z_{jp}^S).w_j$$

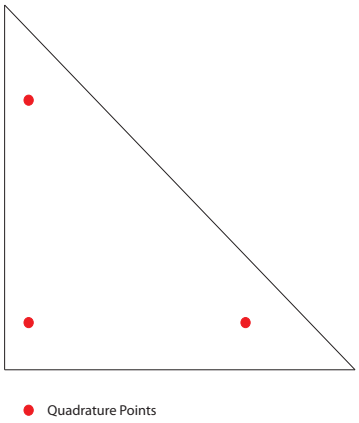


Figure 2.6: Evaluation of Riemann solver on face of tetrahedron

Volume Integral

The volume integral, V_I is defined as

$$V_I = \int_\Omega (F, G, H).\nabla.\phi_l(\xi_p^V(x, y, z), \eta_p^V(x, y, z), \zeta_p^V(x, y, z))d\Omega \tag{2.9.2}$$

or with the grad term expanded

$$V_I = \int_\Omega (F, G, H).\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right).\phi_l(\xi_p^V(x, y, z), \eta_p^V(x, y, z), \zeta_p^V(x, y, z))d\Omega \tag{2.9.3}$$

We can expand this integral further into three distinct integrals, one for each conservative flux F, G, H

$$\begin{aligned}
V_I^F &= \int_\Omega F.\frac{\partial}{\partial x}\phi_l(\xi_p^V(x, y, z), \eta_p^V(x, y, z), \zeta_p^V(x, y, z))d\Omega \\
V_I^G &= \int_\Omega G.\frac{\partial}{\partial y}\phi_l(\xi_p^V(x, y, z), \eta_p^V(x, y, z), \zeta_p^V(x, y, z))d\Omega \\
V_I^H &= \int_\Omega H.\frac{\partial}{\partial z}\phi_l(\xi_p^V(x, y, z), \eta_p^V(x, y, z), \zeta_p^V(x, y, z))d\Omega
\end{aligned}$$

The evaluation of all three integrals is identical, for brevity we will show the method for the F conservative flux integral, and the method is the same for the integrals involving

the G and H conservative fluxes. The above expression for the volume quadrature is evaluated in the same way as the surface integral, however, there is *no* need to derive a transformation similar to (2.9.1). We know the basis functions have been mapped from the physical space to the reference space using the transformation $\chi_{x\xi}$ and the quadrature points are mapped from the reference space to the physical space using $\chi^{-1}_{x\xi}$. This can summarised as:

$$\chi_{x\xi}(\chi^{-1}_{x\xi}) = \xi$$

this means the basis functions are evaluated using the quadrature points $\xi^V_p, \eta^V_p, \zeta^V_p$ in the reference space with no need to map them to the physical space.

The final expression is given as (for the F conservative flux integral)

$$V^F_I = \mathbf{J}_V. \int_{\hat{\Omega}} .F.(\frac{\partial\phi(\xi,\eta,\zeta)}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\phi(\xi,\eta,\zeta)}{\partial\eta}\frac{\partial\eta}{\partial x} + \frac{\partial\phi(\xi,\eta,\zeta)}{\partial\zeta}\frac{\partial\zeta}{\partial x})d\hat{\Omega} \tag{2.9.4}$$

where $\mathbf{J}_V.$ is the Jacobian of the transformation and w_j is weight associated with the quadrature points. $\frac{\partial\xi}{\partial x}, \frac{\partial\eta}{\partial y}, \frac{\partial\zeta}{\partial z}$ are the result of the chain rule used to define the integral in the reference space - these expressions can be derived by inverting the Jacobian[5]. Similar to the quadrature method for the surface integral, this quadrature method requires multiplication by the volume of the reference element, in this case $\frac{1}{6}$. This yields the well-known expression $|V| = \frac{1}{6}\mathbf{J}_V$: the volume of the tetrahedron in the physical space.

$$V^F_I \approx |V| \sum_{j=1}^{J} F.(\frac{\partial\phi(\xi^V_{jp},\eta^V_{jp},\zeta^V_{jp})}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\phi(\xi^V_{jp},\eta^V_{jp},\zeta^V_{jp})}{\partial\eta}\frac{\partial\eta}{\partial x} + \frac{\partial\phi(\xi^V_{jp},\eta^V_{jp},\zeta^V_{jp})}{\partial\zeta}\frac{\partial\zeta}{\partial x})w_j$$

the volume quadratures for the G, H conservative fluxes are calculated in an identical fashion.

$$\begin{aligned} V^G_I &\approx |V| \sum_{j=1}^{J} G.(\frac{\partial\phi(\xi^V_{jp},\eta^V_{jp},\zeta^V_{jp})}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\phi(\xi^V_{jp},\eta^V_{jp},\zeta^V_{jp})}{\partial\eta}\frac{\partial\eta}{\partial y} + \frac{\partial\phi(\xi^V_{jp},\eta^V_{jp},\zeta^V_{jp})}{\partial\zeta}\frac{\partial\zeta}{\partial y})w_j \\ V^H_I &\approx |V| \sum_{j=1}^{J} H.(\frac{\partial\phi(\xi^V_{jp},\eta^V_{jp},\zeta^V_{jp})}{\partial\xi}\frac{\partial\xi}{\partial z} + \frac{\partial\phi(\xi^V_{jp},\eta^V_{jp},\zeta^V_{jp})}{\partial\eta}\frac{\partial\eta}{\partial z} + \frac{\partial\phi(\xi^V_{jp},\eta^V_{jp},\zeta^V_{jp})}{\partial\zeta}\frac{\partial\zeta}{\partial z})w_j \end{aligned}$$

[5]Please consult the appendix

2.10 Riemann Solvers For The Euler Equations

The Euler equations are a systems of non-linear conservation laws, and can be written as:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} + \frac{\partial \mathbf{G}}{\partial \mathbf{x}} + \frac{\partial \mathbf{H}}{\partial \mathbf{x}} = 0 \tag{2.10.1}$$

where ($\mathbf{F}, \mathbf{G}, \mathbf{H}$ are the conservative flux, and $\mathbf{U}$ is the vector of conserved variables. The Riemann problem consists of the hyperbolic conservation form of the Euler equations (2.10.1) with the following initial conditions

$$U(\mathbf{x}, 0) = U^0(\mathbf{x}) = \begin{cases} \mathbf{U}_L & \text{if } x < 0 \\ \mathbf{U}_R & \text{if } x > 0 \end{cases}$$

There arises an important (unknown) region between the extreme left and right characteristic waves for the Riemann solution to the Euler equation called the *Star region*. The primitive variables within region are denoted by a $*$ subscript.

The solution to the above problem yields three different types of waves: *Shocks, contact discontinuity and rarefaction waves*. The middle wave is always a contact discontinuity, and the left and right waves can be either rarefaction or shock waves.

Rarefaction
A *rarefaction* or an expansion wave, is a type of smooth wave across which all the flow variables change, in particular the pressure of the pressure and density *decrease*. The wave speed $u - a$ and $u + a$ increase monotonically from left to right. Rarefaction waves are always composed of acoustic waves - entropy waves do not create rarefaction waves. The last characteristic in the region of the high pressure is called the *head*, and the last characteristic in the region of the low pressure is called the *tail* .

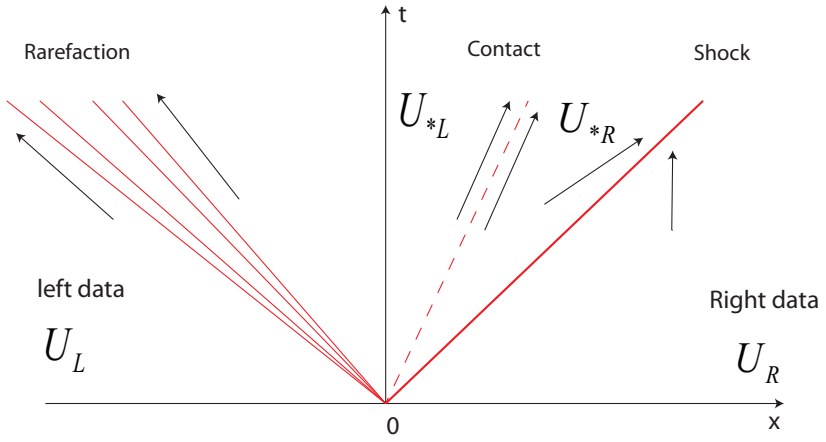
Shock waves
A *Shock wave* or a compression wave is a type of discontinuous wave. Like the rarefaction wave the flow variables across the shock wave change, in particular the pressure and density *increase*. The wave speeds $u - a$ and $u + a$ decrease monotonically from left to right. Unlike the characteristics of the rarefaction wave, which diverge, the characteristics of the compression wave converge forming a shock wave. A shock wave is type of jump discontinuity and are governed by the *Rankine-Hugoniot condition*[6].

contact discontinuity
In complete contrast to the rarefaction and shock waves, who's characteristics must either diverge or converge. The characteristics of a contact discontinuity are parallel to each other. As a result of this, the velocity and pressure don't change across a contact discontinuity, and the contact discontinuities move with the fluid. However, the

[6]This will be discussed in the proceeding section

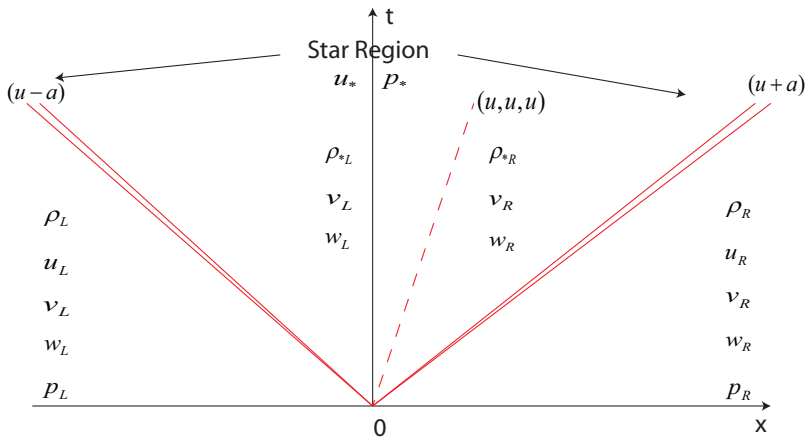
density and anything which is dependent upon the density does. As there is a pressure equilibrium, no gas will flow across a contact discontinuity. Like the Shock wave, the contact discontinuity must obey the Rankine-Hugoniot condition and the entropy may change.

Figure 2.7: 1-D wave structure of the Euler equations (replicated from [68])



This method can be applied in multi-dimensions, by applying it in each direction separately. Below we given an example of the three-dimensional split, and for this particular case (the x direction) the u velocity is unknown in the star region.

Figure 2.8: x direction for the three dimensional split of the Euler equations(replicated from [68])



Writing the Euler equations (2.10.1) in a quasi-linear form

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A}(\mathbf{W})\frac{\partial \mathbf{W}}{\partial x} + \mathbf{B}(\mathbf{W})\frac{\partial \mathbf{W}}{\partial y} + \mathbf{C}(\mathbf{W})\frac{\partial \mathbf{W}}{\partial z} = 0 \tag{2.10.2}$$

we can obtain the eigenvalues λ_i associated with the 5 conserved variables (characteristic variables) for the matrix[7] $\mathbf{A}(\mathbf{W})$ as

$$\lambda_1 = u - a, \quad \lambda_2 = \lambda_3 = \lambda_4 = u, \quad \lambda_5 = u + a$$

to obtain the eigenvalues for the $\mathbf{B}(\mathbf{W})$ and $\mathbf{C}(\mathbf{W})$ matrices we merely replace the u velocity vector with v, w respectively. We can also deduce the corresponding eigenvectors $\mathbf{K}^i$ which correspond to the 3 types of waves. $\mathbf{K}^{2,3,4}$ are contact discontinuities, and $\mathbf{K}^1$ and $\mathbf{K}^5$ are either rarefaction or shock waves. The quasi-linear form of the Euler equation allows us to further analyse the properties of the waves. It can be shown that the Pressure and velocity across as contact discontinuity are constant, it is only the density that changes. In a similar analysis of the rarefaction and shock wave all the primitive variables change. However, the rarefaction is a smooth expansion wave which is bounded by two characteristics called the *head and tail*, and the shock is a compression wave and the characteristic converge. The central wave is always a contact discontinuity, which leaves the possibilities of the rarefaction and shock wave being either left or right waves.

With this information it possible to construct a Riemann solver which is a able to establishing the right side flux.

2.10.1 HLLC Riemann solver

The HLLC solver is an approximate Riemann solver, and is an improvement of the HLL Riemann solver proposed by Harten, Lax Van Leer(1983). The HLL Riemann solver has many appealing features; it is entropy satisfying without the need for an entropy fix, and is relatively simple to program. Unfortunately, due to the single intermediate state of the HLL solvers it is too diffusive and result in the excessive smearing of the contact waves. To overcome this issue the HLL Riemann solver was modified by Toro et.al to account for the lack of the contact wave, and this modified version was called the HLLC Riemann solver, where the *C* in HLLC stands for *contact*.

In the HLLC Riemann solver, the intermediate region is split into two sections.

the star region (intermediate region) is approximated by two waves which account for the contact wave. The HLLC has been widely used in multidimensional problems, and has been seen to been a reliable and robust Riemann solver.

The wave diagram of the solution is defined by four constant states split, separated by three waves. The wave solutions of the Euler equation consist of one contact wave, and two acoustic wave, which can be either shocks or expansion fans[58]

The four states of the Euler equation, moving from left to right, are U_L, U_{*L}, U_{*R}, U_R, where the $*$ subscript represents the two intermediate states, these are sometimes called the star regions, and unlike the HLL solver the HLLC has two star regions. The corresponding waves have speed S_L the speed of the left acoustic wave,S_* the speed of the

[7] **A**, **B** and **C** are defined in the appendix

contact wave, and S_R the speed of the right acoustic wave.

The first step is to calculate the speed of the left (S_L), right(S_R) and the middle wave(S_* - which is always a contact wave). This is given by

$$S_L = u_L - q_L a_L; \quad S_* = U_*; \quad S_R = u_R - q_R a_R$$

where

$$q_L = \begin{cases} 1 & \text{if } p_* \leq p_L \\ \left[1 + \frac{\gamma+1}{2\gamma}\left(\frac{p_*}{p_L - 1}\right)\right]^{1/2} & \text{if } p_* > p_L \end{cases}$$

The expression for q_R is obtained by simply replacing L with R. In the above expression the value of the pressure in the star region, p_*, is required. This can calculated using a wave speed or pressure-based estimation.

Next we calculate the state variables $U_{L,R,*}$. calculating U_L and U_R is simple. However, to calculate the state variables in the star region, we use

$$U_{*L} = \rho_L\left(\frac{S_L - U_L}{S_L - S_*}\right)\begin{bmatrix} 1 \\ S_* \\ V_L \\ W_L \\ \frac{E_L}{\rho_L} + (S_* - U_L)\left[S_* + \frac{p_L}{\rho_L(S_L - U_L)}\right] \end{bmatrix}$$

once again, to calculate the right star region value, we replace the L, with an R. Now that we have the estimation for the speed of the waves, and established the values of the state variables in all the regions it is possible to calculate the fluxes using the expression below.

$$H_{HLLC} = \begin{cases} F(U_L) & \text{if } S_L > 0 \\ F(U_L) + S_L(U_{*L} - U_L) & \text{if } S_L \leq 0 < S_* \\ F(U_R) + S_R(U_{*R} - U_R) & \text{if } S_* \leq 0 \leq S_R \\ F(U_L) & \text{if } S_R < 0 \end{cases}$$

Grid Alignment of the conserved variables

Unlike the sturctured grid where the conserved variables are aligned with the outward normals of the face $\hat{\mathbf{n}} = (\hat{n_x}, \hat{n_y}, \hat{n_z})$, on an unstructured grid this is not the case. As a result the vector of conserved variables $\mathbf{U}$ have to be aligned with the normals of the face. To derive the vector of conserved variables which are multiple by the rotational matrix $\mathbf{T}$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & cos\theta sin\phi & sin\theta sin\phi & cos\theta & 0 \\ 0 & -sin\theta & cos\theta & 0 & 0 \\ 0 & cos\theta cos\phi & sin\theta cos\phi & -sin\phi & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus, we get the vector of conserved variables aligned with the outward normal on a face of the tetrahedron.

$$\hat{\mathbf{U}} = \mathbf{T}_k \mathbf{U}$$

Where the subscript k refers to the row of the rotational matix $\mathbf{T}$. The inverse of the rotational matrix is

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & cos\theta sin\phi & -sin\theta & cos\theta cos\phi & 0 \\ 0 & -sin\theta sin\phi & cos\theta & sin\theta cos\phi & 0 \\ 0 & cos\theta & 0 & -sin\phi & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

After calculating the conserved variable in the rotated cooridnate system (x' ,y',z') we can rotate back to the coordinate system we orginally defined the conserved variables in (x, y, z) by

$$\hat{\mathbf{F}}.\hat{\mathbf{n}} = \hat{F}_k \hat{n_{xk}} + \hat{G}_k \hat{n_{yk}} + \hat{H}_k \hat{n_{zk}} = \mathbf{T}_k^{-1} \hat{\mathbf{F}}(\mathbf{T}_k \mathbf{U})$$

Figure 2.9 illustrates the rotation of the coordinate system. The red axis (x' ,y',z') denote the coordinate axes for the system which points in the outward normal of face, whereas (x, y, z) is the cooridnate which is not alligned with the outward normal. L,R refer to the left and right hand side of the face.

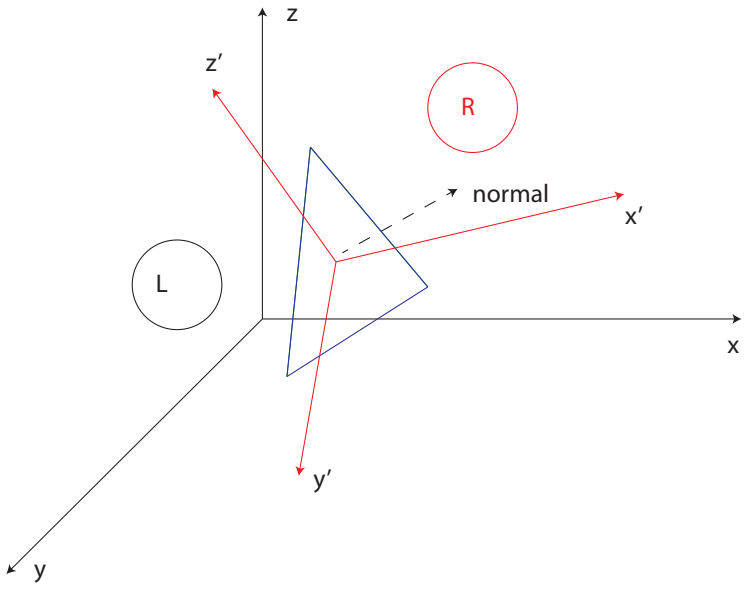


Figure 2.9: Rotation of the coordinate system to align with the outward normals

2.11 Runge-Kutta method

To solve the semi-discrete form the most popular approach is the Runge-Kutta method, which is used to progress the solution through time. The TVD Runge-Kutta scheme developed by Shu and Osher[61] can theoretically achieve a k^{th} order of accuracy. However, due to memory storage issues upto fourth order schemes are generally used. The CFL(Courant-Friedrichs-Lewy) number is used to calculate the size of a permissible time step Δt. It is important the size of the timestep is not too large as this would render the numerical scheme unstable.

For discontinuous Galerkin the following equation provides a good approximation for the CFL number

$$CFL = \frac{1}{2k+1} \quad k = 0,1,..P$$

from which we can derive the size of the timestep Δt. For the 1D linear advection equation this is given by

$$\Delta t = \frac{CFL.\Delta x}{C}$$

where C is the advection velocity, and Δx is the size of the cell. For the Euler equations on an unstructured grid, the size of the time step is calculated using a similar approach.

$$\Delta t = \frac{CFL.\Delta x}{max|\lambda^p|}$$

The maximum value of the eigenvalue λ^p is given by $max(u, v, w) + a$, where a is the speed of sound. To balance the dimensions of the above equation, we have to determine a 1D parametre for the size of a 3D element. The most natural choice is to determine a radius. To do this we inscribe the element inside a sphere, and calculate the radius of the sphere[28].

Cockburn and Shu[13] document the CFL numbers for the increasing order of spatial accuracy for increasing order of the temporal accuracy. From there studies it has been shown it is possible to construct very high-order methods (upto 9^{th} order) which can be solved by either 3^{rd} or 4^{th} order Runge-Kutta methods.

The 2nd and 3rd order TVD Runge-Kutta method as given in[24] will be used throughout this thesis.

Although the Runge-Kutta method is the most prevalent of all the time integration methods, researchers have experimented with other method including the Lax-Wandroff[55]. Unfortunately this method is more complicated to program than then the Runge-Kutta method.

2.12 Limiting methods for high-order methods

Arguably one of the most important theorems in the work of modelling and simulation is that of Godunov, which[38] describes as:

Linear monotonicity-preserving methods are first order accurate at best

Simply put, this means that unlimited high-order schemes (2^{nd} order and greater) which try to model flows which posses discontinuities or shocks will be characterised by the following effects.

- Away from the discontinuity the first order accuracy starts to diminish.
- At the discontinuity the convergence of the solution is destroyed.
- The scheme will exhibit oscillations at the discontinuity which are known as the *Gibbs Oscillations*.

these are collectively known as the ***Gibbs phenomenon***, an in depth analysis can be found in[23].

To counter the problem posed as a result of the Godunov's theorem, many methods have been developed, including artificial viscosity methods, TVD methods and (W)ENO methods, which are part of the larger class of methods called *high-resolution schemes* - these are categorised by the following properties:

- At least 2^{nd} order accuracy in the smooth region of the flow
- Free from Gibbs Oscillations at the discontinuity

A popular way of implementing high-resolution schemes work is by dropping the numerical scheme to 1^{st} order accuracy at the discontinuity to avoid the Gibb's oscillations, and retain the higher orders of accuracy in the smooth regions where the Gibbs oscillations do not occur. This process is carried out by *slope limiters* or *flux limiters*, both aim to detect cells which may cause the development of these oscillations, and thus drop the order of accuracy to 1^{st} order if there is a steep gradient. The MUSCL type schemes which are sometimes known as reconstruction-evolution methods rely upon the slope limiter approach, whereby the slopes of the reconstructed values are restricted to be within the cell averages. A historical overview is given in[40], with analysis and introduction to the above method given in[17].

In the context of discontinuous Galekrin the first slope limiter was developed by *Cockburn and Shu*[10] for a 1D case, which was extended and applied to a 2D case, for both structured and unstructured domains[12]. A 3D limiter *inspired* by the aforementioned limiter developed by *Cockburn and Shu* was developed and applied in[58]

upto 3^{rd} order accuracy. The limiter of CockBurn and Shu is Total-Variation Bounded (TVB) and is at most 3^{rd} order accurate.

Alternative approaches to the design of limiters for DG have been proposed by *Qiu and Shu*[54] who developed a limiter on structured grids using a WENO approach, this method was extended to unstructured meshes by*Zhu et.al*[76] for 2D domains. A 3D polynomial type Hermite-WENO limiter has also been constructed[47]. The WENO approach to limiting is diffcult and computational expension, especially in high-dimensions. It introduces back into the development of DG what DG was meant to eradicate, namely the need for large stencils.

In this thesis we will be utilising a TVD limiter which is the common to both FVM and DG, namely the $Barth - Jespersen$. The concept of total variation diminishing TVD was introduced by *Harten*[26], the fundamental idea behind the TVD approach is to ensure that maxima are non-increasing, minima non-decreasing and no new local extrema is formed. The total variation of a function for discrete cases is defined as

$$TV(u) = \sum_i |u_{i+1} - u_i|$$

Assuming the current and previous time levels $n + 1$ and n, respectively, then the TVD must obey the following property

$$TV(u^{n+1}) \leq TV(u^n) \tag{2.12.1}$$

TVD limiters are non linear functions which circumvent the Godunov theorem, which applies only to linear schemes. As a consequence, for flows which exhibit shocks, the application of a non-linear which ensures the TVD property(2.12.1) prevents the introduction of the Gibb's oscillations.

2.12.1 Barth-Jespersen Limiter

The above limiters mentioned for DG are difficult to implement in 3D. Probably one of the most robust type of limiters is proposed by Barth and Jespersen[3] on unstructured grids. It is simpler to implement than the DG limiters mentioned previously, but more importantly guarantees the consistency required to conduct a comparison between the FVM and DG methods. For 2nd order accuracy the Barth-Jespersen limiter for FVM and DG are implemented in a similar manner; information from only the direct side neighbours is required . For 3rd order accuracy FVM requires the derivation of higher order derivatives for the reconstruction expansion[48].

The Barth-Jespersen limiter is based on the principle of restricting the reconstructed solution $U_h(x_j)$ at each quadrature point j between the maximum and minimum cell averages of the neighbouring cells. For each cell there are 4 neighbours and the cell we consider, therefore, 5 elements in total(Figure 2.12.1).

K_0 is the cell upon which we enforcing the limiting and $K = 1, 4$ are the neighbouring cells. then we establish the maximum and minimum

$$\begin{aligned} \bar{U}^{max} &= max(\bar{U}_K | \forall K = 0, 4) \\ \bar{U}^{min} &= min(\bar{U}_K | \forall K = 0, 4) \end{aligned}$$

Where the cell averages are given by

$$\bar{U}_K = \frac{1}{|V|} \int_\Omega U_h d\Omega = \hat{u}_{0K} \quad \text{for} \quad K = 0, 4$$

Where $\hat{u}_{0K}$ is the 1st order degree of freedom in element K. We require the reconstructed solution to be between,

$$\bar{U}^{min} \leq U_h(x_j) \leq \bar{U}^{max}$$

To ensure the monotonicity preserving aspect of the limiter. If this inequality is violated then it is assumed a steep gradient is present in the solution, and the solution is limited.

The Barth-Jespersen method works upon the approach of finding the largest possible ψ_j at the quadrature points j such that local extrema are prevented from forming.

The limiting procedure can summerised in the following three steps

1. Find the largest negative $(\bar{U}^{min} - U_0)$ and positive gradients $(\bar{U}^{max} - U_0)$ with respect to the target element K_0 and the direct side neighbour elements $K_{1,2,3,4}$. Where $\bar{U}^{min}$ and $\bar{U}^{max}$ are the minimum and maximum cell averages of any of the neighbouring elements.

2. Calculate the reconstuction values at the quadratures points on the faces of the target element K_0.
$$U_h = \sum_{l=0}^{Np-1} \hat{u}_l(t) \phi_l(\xi(\mathbf{x}), \eta(\mathbf{x}), \zeta(\mathbf{x}))$$

3. Based upon the previous two steps we can calculate the value of ψ_j.

$$\psi_j = \begin{cases} min(1, \frac{\bar{U}^{max} - U_0}{U(x_j) - U_0}) & \text{if } U(x_j) - U_0 > 0 \\ min(1, \frac{\bar{U}^{min} - U_0}{U(x_j) - U_0}) & \text{if } U(x_j) - U_0 < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\psi = min(\psi_j)$$

Once the value of ψ_j has been obtained at each quadrature point we can apply this to obtain an updated reconstruction. Where ϕ_l are the basis functions for the corresponding degrees of freedom.

$$U_h = \hat{u}_0 + \psi[\hat{u}_1\phi_1 + \hat{u}_2\phi_2 + \hat{u}_3\phi_3]$$

As we can see from the definition of ψ_j if $\psi_j = 1$ then we obtained the original unlimited value on the face.

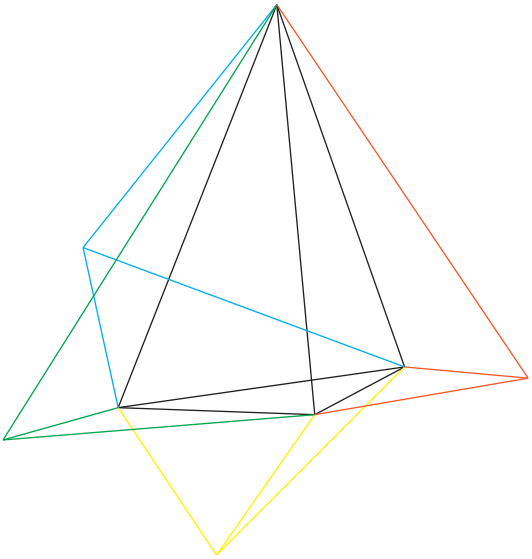


Figure 2.10: Target element and neighbours for limiting method 3D

CHAPTER 3

Validation of the DG Solver

The size and complexity of the three-dimensional discontinuous Galerkin code requires that a number of test cases be conducted in order to ensure the chances of accumulating erroneous procedures is minimised.

The basis functions are required in the evaluation of the initialisation of the flow, the surface fluxes, and the reconstruction of the solution back to the physical space. Whereas the partial derivatives of the basis functions are required in the volume fluxes. Therefore, it is essential that the basis functions and partial derivatives of the basis functions are evaluated properly. The Basis functions are evaluated using both the surface and volume quadrature points, hence, it is just as important to validate the quadrature points.

As will be shown, the simulation of the constant initial condition C validates the most aspects of the code, and is important in the context of the discontinuous Galerkin method.

3.1 Validation of the basis functions and norms

Orthogonal basis functions are used in the discontinuous Galerkin, which results in a diagonal mass matrix which can be pre-processed before the start of the simulation.

Orthogonal basis functions, or to be more precise orthogonal polynomials obey the following result

$$\int \phi_p \phi_q d\hat{\Omega} = a_{pq} \delta_{pq}$$

where the Kronecker delta δ_{pq} is:

if $p = q$

$$\delta_{pq} = 1$$

if $p \neq q$

$$\delta_{pq} = 0$$

which translate into a diagonal mass matrix

is given by the formula

$$\int \phi_p \phi_q d\hat{\Omega} = \begin{pmatrix} a_{1,1} & 0 & 0 & . & . & 0 \\ 0 & a_{2,2} & 0 & . & . & 0 \\ . & . & a_{3,3} & . & . & 0 \\ . & . & . & . & . & 0 \\ . & . & . & . & a_{n-1,n-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{n,n} \end{pmatrix}$$

We can validate the basis functions by writing a double loop such that the first loop is carried out over the basis functions p and the second loop is carried out over q. When $p = q$ then a non-zero answer should be returned, and subsequently when $p \neq q$ an answer of zero should be returned. If this procedure is carried out using a quadrature method, then in part it also validates the volume quadrature points too.

Validating the norms using a quadrature approximation, this becomes

$$\int \phi_p \phi_q d\hat{\Omega} = \frac{1}{6}. \sum_{j=0}^{J} \phi_p(x_j, y_j, z_j) \phi_p(x_j, y_j, z_j).w_j$$

```
 do p = 1,NP       (p basis functions)

   do q = 1,NP     (q basis functions)

  a(p,q) =0.0      (initialising loop)

     do j = 1,J    (loop over the quadrature points)

  a(p,q) = a(p,q) + Basis(x(j),y(j),z(j),p)*Basis(x(j),y(j),z(j),q)*w(j)

     end do
  end do

 a(p) = a(p)/6.0

end do
```

The norms for the basis functions up to the 2^{nd} order of accuracy are given in the appendix.

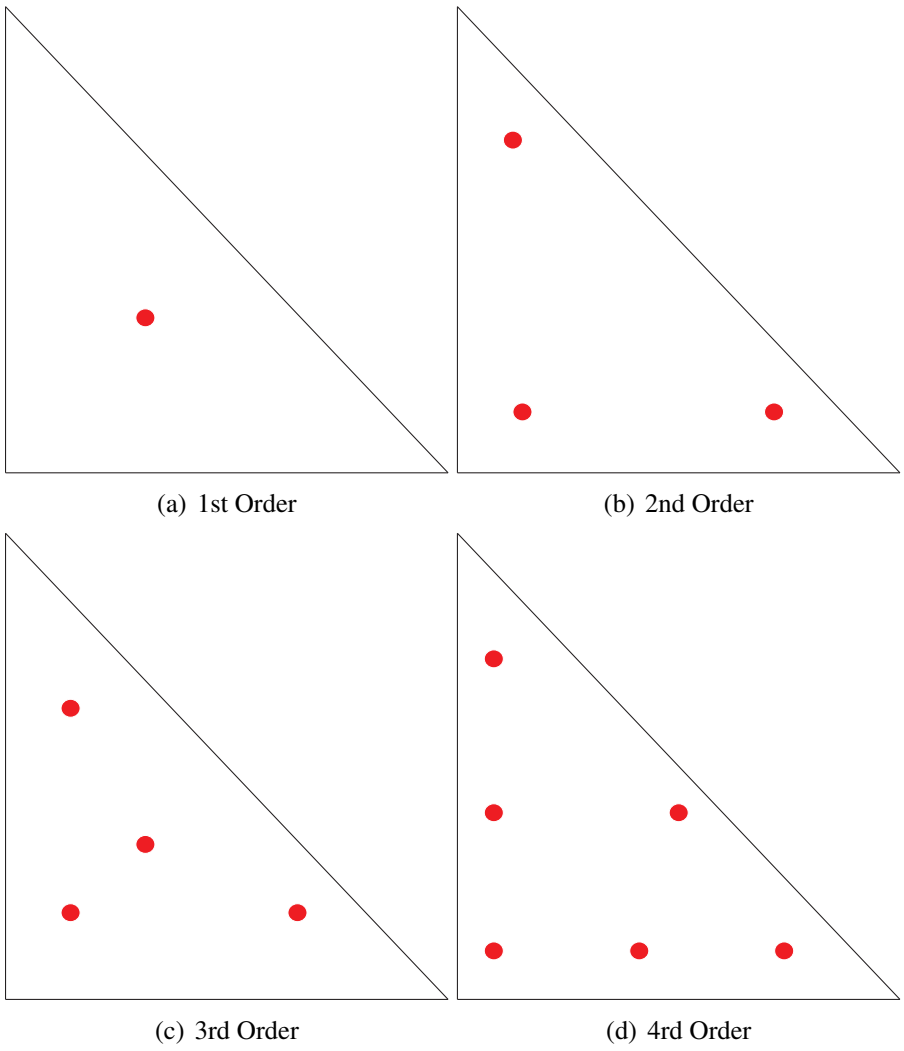
3.2 Validation of quadrature points

3.2.1 Surface Quadrature

The surface quadrature points where validated using the testcase defined in[56]. Whereby the order of the quadrature method, hence the number of quadratures points on the triangle is increased. This test

$$\int_0^1 \int_0^{1-x} \sqrt{x+y}\, dxdy = 0.4$$

Is performed on a unit reference tetrahedron with dimensions $[0,1]^2$, therefore, no transformation is required to the reference element and only the quadrature points are being tested.



(a) 1st Order (b) 2nd Order

(c) 3rd Order (d) 4rd Order

Figure 3.1: Example of distribution of quadrature points on reference triangle $[0,1]^2$ for upto 4th order accuracy

In **Figure 3.1** the distribution of quadrature points is illustrated for increasing order of accuracy. The quadrature points are calculated in[19], and are stated in the appendix.

Figure 3.2: Validation of the Surface Quadrature points

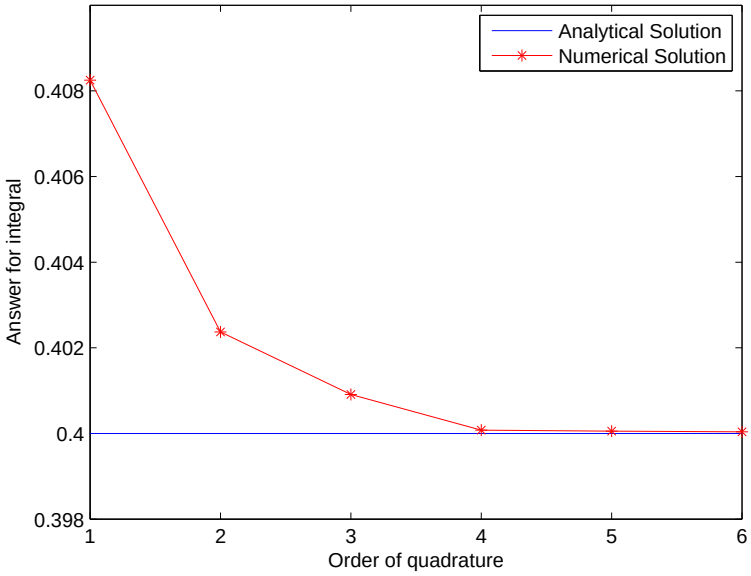


Table 3.1: Results for Surface quadrature validation Testcase

Order	Answer	Absolute error
1	0.408248	8.2483E-3
2	0.402369	2.3689E-3
3	0.400910	9.0979E-4
4	0.400078	7.8439E-5
5	0.400053	5.17431E-5
6	0.400039	3.92430E-5

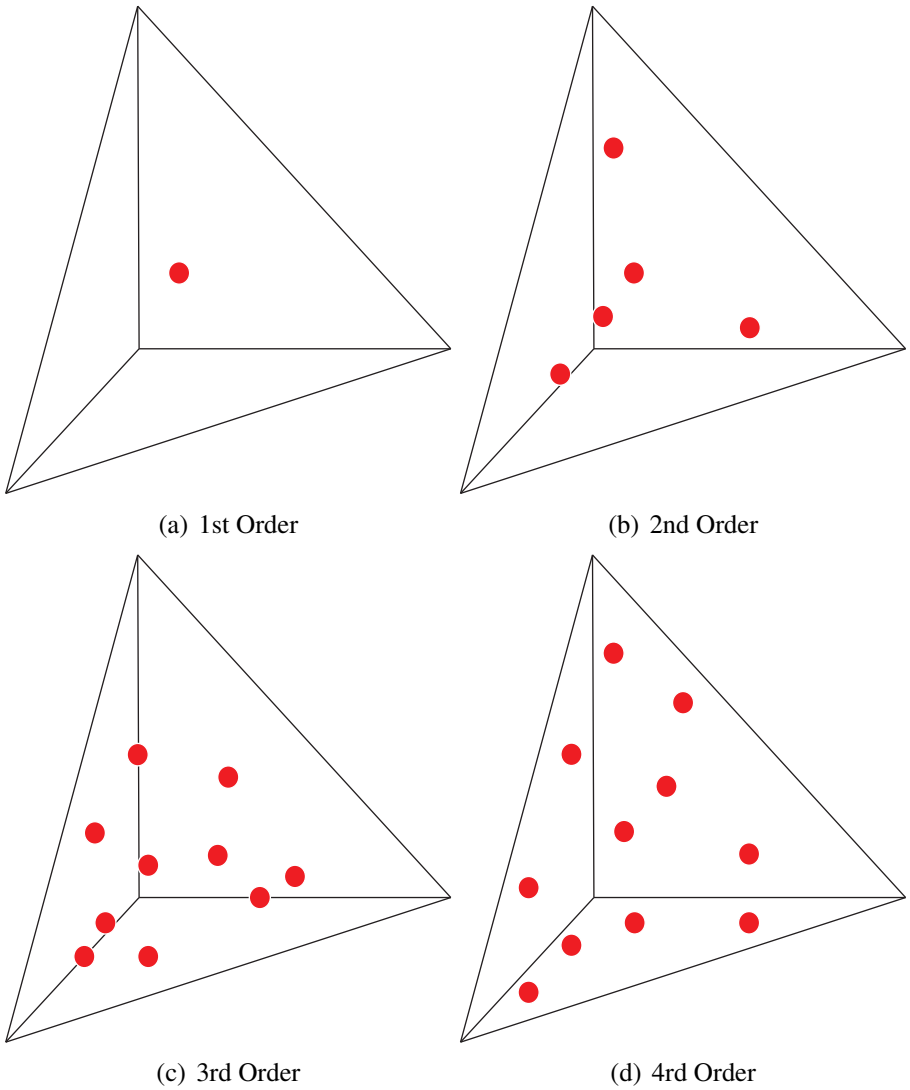
Figure 3.2 shows the convergence of the numerical solution to the analytical solution with increasing quadrature order. An absolute error is given in **Table 3.1**.

3.2.2 Volume Quadrature

The volume quadrature points are validated using the testcase defined in[50].

$$\int_0^1 \int_0^{1-x} \int_0^{1-x-y} \sqrt{x+y+z}\, dxdydz = 0.142857142$$

A similar approach for the analysis of the triangle quadrature points is used for the tetrahedron. The above testcase is solved for increased quadrature points on a unit reference tetrahedron in $[0,1]^3$.



(a) 1st Order (b) 2nd Order

(c) 3rd Order (d) 4rd Order

Figure 3.3: Example of distribution of quadrature points on reference tetrahedron $[0, 1]^3$ for upto 4th order accuracy

In **Figure 3.3** the distribution of quadrature points on a unit tetrahedron is illustrated. The quadrature points are calculated in[34]

Figure 3.4: Validation of the Volume Quadrature points

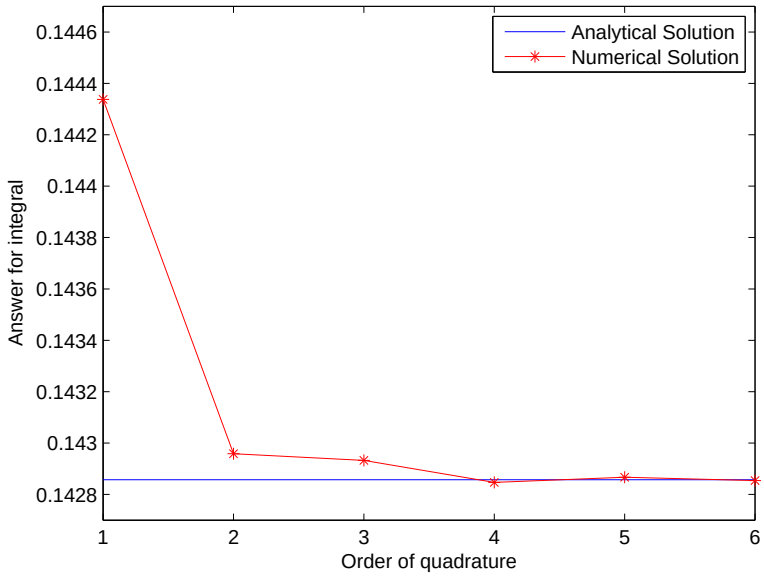


Table 3.2: Results for Volume quadrature validation Testcase

Order	Answer	Absolute error
1	0.1443376	0.001480455
2	0.1429589	0.00010174513
3	0.1429328	0.000075608492
4	0.1428471	0.000010088086
5	0.1428669	0.0000097751617
6	0.1428542	0.0000029802322

3.3 Validation of high-order surface and volume fluxes

Consider the constant initial condition C

$$U_0(x, y, z) = C$$

then the following theory can be postulated

Theorem 1 *If $U_0(x, y, z) = C$ then the first-order degree of freedom is $\hat{u}_0 = C$. Subsequently, the higher-order degrees of freedom are $\hat{u}_l = 0, \forall\ l = 1, 2, ..Np$*

Proof

Initialisation of the first-order degree of freedom

The first-order initialisation of the constant initial condition C is:

$$\hat{u}_0 = \frac{1}{a_0} \int C.\phi_0(x, y, z) d\hat{\Omega}$$

this can be simplified to

$$\hat{u}_0 = 6C \int 1.\phi_0(x, y, z) d\hat{\Omega}$$

where $a_0 = \frac{1}{6}$. The 1 multiplying the basis function has been left in purposefully. The first-order basis function is 1, therefore, the integral is the first entry of the mass matrix i.e

$$\int 1.1 d\hat{\Omega} = \frac{1}{6}$$

which gives the area of the reference tetrahedron. The initialisation integral becomes

$$\hat{u}_0 = 6.C.\frac{1}{6} = C$$

This proves that the first-order degree of freedom will give the constant initial condition C

Initialisation of the higher-order degrees of freedom

For the higher-order basis functions

$$\hat{u}_l = \frac{1}{a_l} \int C.\phi_l(x, y, z) d\hat{\Omega}$$

where l=1,NP

factorsing out the constant C, this integral becomes

$$\hat{u}_l = \frac{C}{a_l} \int 1.\phi_l(x, y, z) d\hat{\Omega}$$

the 1 here is merely the first basis function, therefore, this can be written as

$$\hat{u}_l = \frac{C}{a_l} \int \phi_0(x, y, z).\phi_l(x, y, z) d\hat{\Omega}$$

as the basis functions are orthogonal i.e

$$\int \phi_p \phi_q d\hat{\Omega} = a_{pq} \delta_{pq}$$

this results in

$$\hat{u}_l = 0; l = 1, NP$$

In summary, the set of degrees of freedom for the constant initial condition C is

$$\{C, 0, 0, ..0, |\forall l = 0, NP\}$$

□

Now that it has been proved that for the initial condition C,the high-order degrees of freedom are 0, we can employ this fact to validate the surface and volume quadratures for higher-orders of accuracy. The reconstruction of the solution at the quadrature points is given by

$$U_h = \sum_{l=1}^{Np} \hat{u}_l \phi_l(\xi, \eta, \zeta)$$

this reduces the reconstruction for the initial condition C to:

$$U_h = C$$

recalling the discontinuous Galerkin formulation

$$\frac{d\hat{u}_l}{dt} = \frac{1}{a_l} \Big(\int_\Omega \nabla.\phi(\xi, \eta, \zeta).F(U_h) - \sum_{m=1}^{4} \int_{\partial\Omega} H(U_h^L, U_h^R, \hat{n}) \phi(\xi, \eta, \zeta) \Big)$$

setting the advection velocity $C_x = C_y = C_z = 1$, then the reconstruction of the solution for the volume quadrature is $F(U_h) = C$. Similarly, the reconstruction of the left and right state for the numerical flux H is $U_h^L = U_h^R = C$. A well known property of the numerical flux $H(U_h^L, U_h^L, \hat{n}) = U_h^L$ or equally $H(U_h^R, U_h^R, \hat{n}) = U_h^R$, resulting in the in $H(U_h^L, U_h^R, \hat{n}) = C$. This reduces the Right-hand side of the Discontinuous Galerkin formulation to

$$\frac{C}{a_l} \Big(\int_\Omega \nabla.\phi(\xi, \eta, \zeta) - \sum_{m=1}^{4} \int_{\partial\Omega} \phi(\xi, \eta, \zeta) \Big)$$

Inside the Parentheses the integrals form the well known divergence theorem, with the volume integral being equal to the surface integral.

$$\int_\Omega \nabla.\phi(\xi, \eta, \zeta) = \sum_{m=1}^{4} \int_{\partial\Omega} \phi(\xi, \eta, \zeta)$$

or

$$\int_{\Omega} \nabla.\phi(\xi,\eta,\zeta) - \sum_{m=1}^{4} \int_{\partial\Omega} \phi(\xi,\eta,\zeta) = 0$$

As a result, for the constant initial condition C, the right-hand side is zero. This process also validates the derivatives of the basis functions.

3.4 Summary of validation procedure

In this chapter it has been shown how to validate the Discontinuous Galerkin formulation

- The basis functions and norms of the basis functions are validated by the construction of a stand-alone procedure which loops over all the basis functions to construct the mass matrix. A diagonal mass matrix should be expected, with non-zero entries on the diagonal only, and zero entries everywhere else.
- The quadrature points for both the surface and volume quadratures are validated using a stand-alone procedure, which solves the problems identified by Rathod et.al, and convergence to the analytical solution is expected with increased order of accuracy of the quadrature points.
- For high-orders of accuracy, there is a contribution from the volume quadratures. To validate both the surface and volume quadratures at orders of accuracy greater than 1, a constant initial condition is used which results in the divergence theorem on the right-hand side , and therefore the right-hand side is zero should be zero. As the basis function have been validated, the partial derivatives in the volume quadrature will also be validated.

CHAPTER 4

3D testcases

In this chapter testcases are analysed for the robustness and accuracy of the 3D discontinuous Galerkin method against the Finite Volume Method for upto 3rd order spatial accuracy without a limiter and 2nd order spatial accuracy when a Barth-Jespersen TVD limiter is applied. A comparison is drawn between the L_0, L_1 errors and the subsequent convergence to the nominal order of accuracy.

The first set of testcases concentrate on the advection equation with initial conditions which are sufficiently smooth, hence we do not expect Gibb's oscillations to form, and there is no need for a limiter. The initial conditions are: $sin(2\pi x)sin(2\pi y)sin(2\pi z)$, $sin(4\pi x)sin(4\pi y)sin(4\pi z)$, $sin(6\pi x)sin(6\pi y)sin(6\pi z)$. These initial conditions ensures that flow is truly three dimensional, and also, as the advection equation can be analytically solved, an error analysis can be conducted with regards to an analytical solution. Although, the aforementioned testcases do not require a limiter to achieve the desired order of convergence, a separate set of testcases will investigate the effect of a Barth-Jespersen TVD limiter applied to the above initial conditions, this is to assess the effect of a limiter on a sufficiently smooth profile.

Extending the single component hyperbolic case (advection equation) to the systems case (Euler equations), a Method of Manufactured Solutions (MMS) is used. This ensures the Euler equations can be analytically solved and also that the solution does not produce any oscillations, hence ensuring a convergence study can be conducted.

Another area of importance is the use of the TVD Runge-Kutta method as opposed to the non-TVD classical Runge-Kutta methods. A discontinuous initial condition is chosen for the advection equation and the errors of the DG2 method are compared when a TVD and non-TVD Runge-Kutta methods are are used.

Further to the smooth testcases mentioned above, and the advection equation with a discontinuous initial condition, the FVM and DG methods are applied to the Euler equations for flows which exhibit discontinuous solutions, namely the Sod's shock-tube case and the implosion and explosion testcases. The implosion case is used as a verification testcase, because of the merging of the shockwaves during the implosion part of the simulation and the subsequent explosion - this is an excellent testcase for the robustness of the solver.

4.1 Smooth Testcases

4.1.1 Linear Advection Equation

The linear advection equation, which is the simplest type of hyperbolic conservation law:

$$\frac{\partial U}{\partial t} = C_x \frac{\partial U}{\partial x} + C_y \frac{\partial U}{\partial y} + C_z \frac{\partial U}{\partial z}$$

With the advection velocities set to $C_x = C_y = C_z = 1$

The flow is initialised using a triple sine wave for three different initial conditions

$$U_0 = U(x, y, z, 0) = sin(W_n \pi x) sin(W_n \pi y) sin(W_n \pi z)$$

For the wavenumbers $W_n = 2, 4, 6$ with periodic boundary conditions enforced in all directions, and the domain size is $[0, 1]^3$.

This problem is solved on four grids (**Table 4.1**), and the L_0 and L_1 errors are calculated in order to determine the nominal order of accuracy. The simulation is run for $T = 1$ coupled with the periodic boundary conditions, this ensures the sine wave initial profile completes one period, returning to the origin of the simulation.

Table 4.1: Grid information for advection equation testcases

	N_G	N_E	2^{nd} order: N_{DOF}	3^{rd} order: N_{DOF}
Grid 1	5 x 5 x 5	662	2648	6620
Grid 2	10 x 10 x 10	5957	23828	59570
Grid 3	20 x 20 x 20	40398	161592	403980
Grid 4	40 x 40 x 40	193836	775344	1938360

Table 4.1, also shows the total number of degrees of freedom (N_{DOF}) for the 2nd and 3rd order DG method, along with the Number of grid points (N_G) and number of elements (N_E). For 1st order DG, N_{DOF} is the same as the number of elements. And for FVM, the N_{DOF} does not change as the order of accuracy increases - FVM has one degree of freedom per element per variable.

The order of the temporal accuracy and CFL number with respect to the spatial accuracy is set according to the findings of[12],these have been replicated and are available in the appendix. For 2nd order accuracy, we use a 2nd order Runge-Kutta method, for 3rd order a 3rd order Runge-kutta method, the CFL number was approximated using the condition

$$CFL \approx \frac{1}{2l + 1}$$

Where l is the order of the polynomial.

Initial Condition: Sin(2πx)Sin(2πy)Sin(2πz)

Tables 4.2, 4.3 give the L_0 and L_1 errors for the advection equation which was initialised with the condition

$$U_0 = U(x, y, z, 0) = sin(2\pi x)sin(2\pi y)sin(2\pi z)$$

This was simulation was run for T=1 physical time

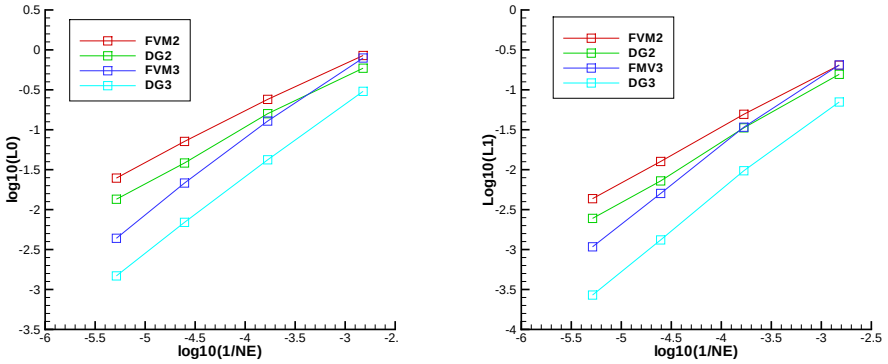
Table 4.2: Numerical Convergence Rates for FVM and DG for linear advection case at time T=1 with IC Sin(2πx)Sin(2πy)Sin(2πz) for P_1 with CFL=0.3

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	662	8.50E-1	2.03E-1			5.90E-1	1.56E-1		
10	5957	2.40E-1	4.94E-2	1.73	1.93	1.59E-1	3.34E-2	1.79	2.11
20	40398	7.14E-2	1.27E-2	1.90	2.13	3.84E-2	7.23E-3	2.23	2.40
40	193836	2.48E-2	4.34E-3	2.02	2.05	1.36E-2	2.45E-3	1.99	2.07

Table 4.3: Numerical Convergence Rates for FVM and DG for linear advection case at time T=1 with IC Sin(2πx)Sin(2πy)Sin(2πz) for P_2 with CFL=0.2

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	662	7.91E-1	2.05E-1			3.03E-1	7.04E-2		
10	5957	1.28E-1	3.40E-2	2.48	2.45	4.20E-2	9.69E-3	2.69	2.71
20	40398	2.15E-2	5.04E-3	2.80	2.99	6.93E-3	1.32E-3	2.83	3.12
40	193836	4.38E-3	1.08E-3	3.05	2.94	1.47E-3	2.71E-4	2.96	3.03

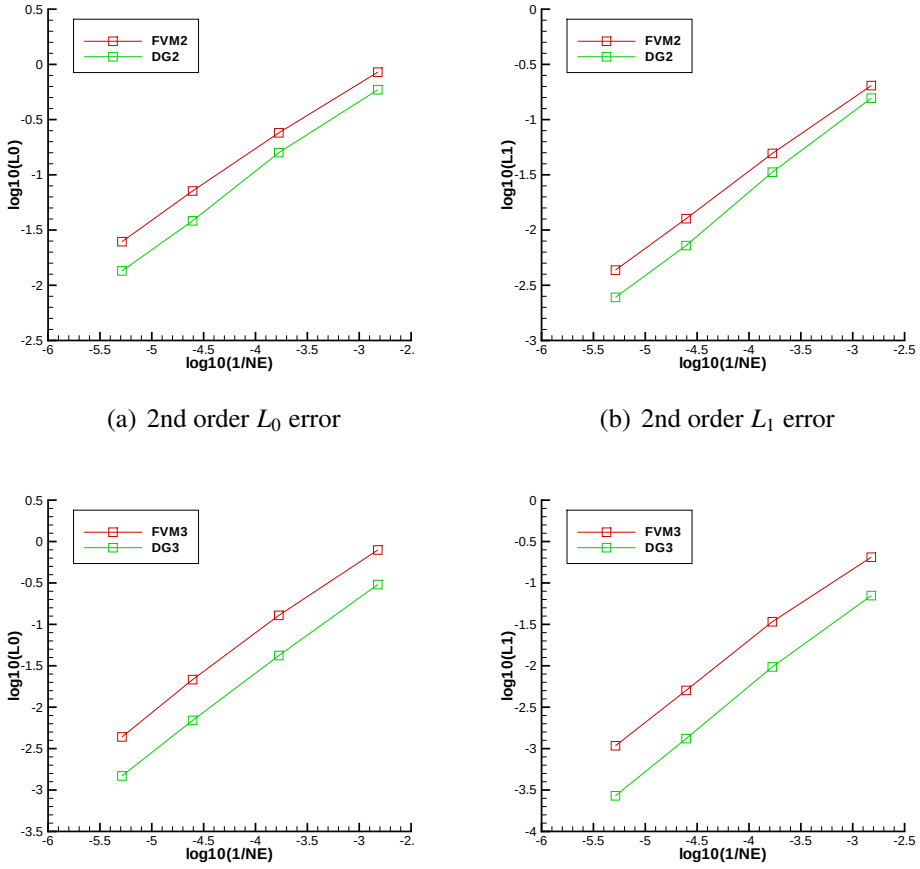
A comparative illustration is given in **Figure 4.1**, this shows the superiority of the DG3 method as opposed to the 2nd order methods.



(a) L0 error 2nd and 3rd order (b) L1 error 2nd and 3rd order

Figure 4.1: Comparison of L_0 and L_1 errors for FVM and DG, with initial conditions Sin(2πx)Sin(2πy)Sin(2πz) at T=1

Figure 4.2 illustrates the comparative convergence of the testcase for L_0 and L_1 for 2nd order and 3rd order accuracies, this makes it easier to notice both the linear convergence as is expected in the logarithmic scale for the nominal order of accuracy and also the superiority of the DG method for both 2nd and 3rd order accuracies.



(a) 2nd order L_0 error

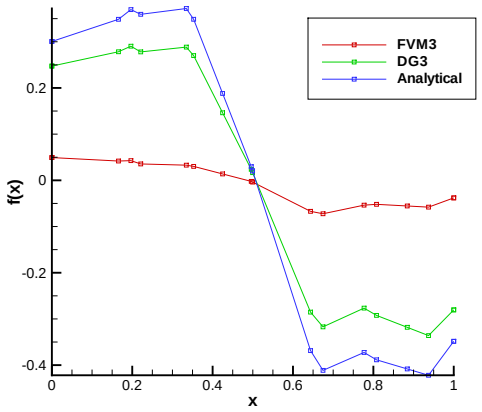
(b) 2nd order L_1 error

(c) 3rd order L_0 error

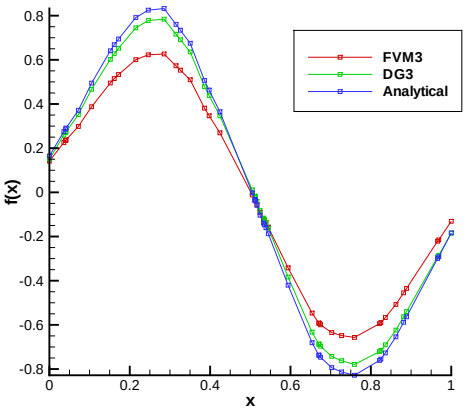
(d) 3nd order L_1 error

Figure 4.2: L_0 and L_1 errors for FVM and DG with initial conditions Sin(2πx)Sin(2πy)Sin(2πz) at T=1

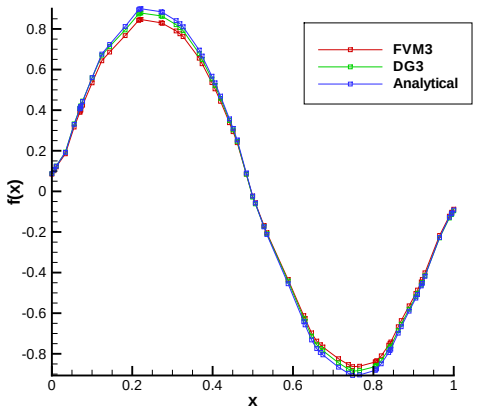
Taking a cut of the $sin(2\pi x)sin(2\pi y)sin(2\pi z)$ isosurfaces at $T = 1$ at positions $z = 0.25$ and $y = 0.25$ we can extract the sine wave propagating in the x direction, which we have termed $f(x) = sin(2\pi x)$. Equally extracting $f(y) = sin(2\pi y)$ or $f(z) = sin(2\pi z)$ would have yielded the sine wave propagating in the y and z directions respectively. A graphical illustration of $f(x)$ is given in **Figure 4.3** for the four grids used in this testcase(**Table 4.1**).



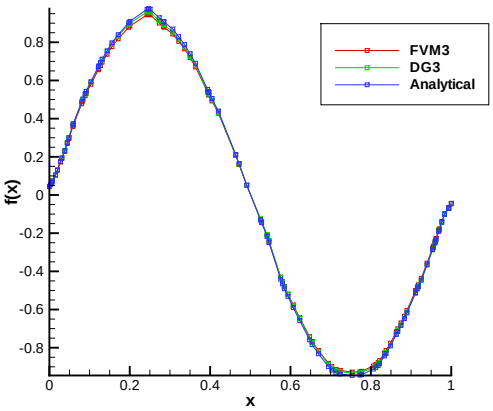
(a) 5 x 5 x 5



(b) 10 x 10 x 10



(c) 20 x 20 x 20



(d) 40 x 40 x 40

Figure 4.3: Comparison of FVM3 and DG3 for profile $f(x) = Sin(2\pi x)$ at T=1

Initial Condition: Sin(4πx)Sin(4πy)Sin(4πz)

Tables 4.4, 4.5 give the L_0 and L_1 errors for the advection equation which was initialised with the condition

$$U_0 = U(x, y, z, 0) = sin(4\pi x)sin(4\pi y)sin(4\pi z)$$

This was simulation was run for T=1 physical time

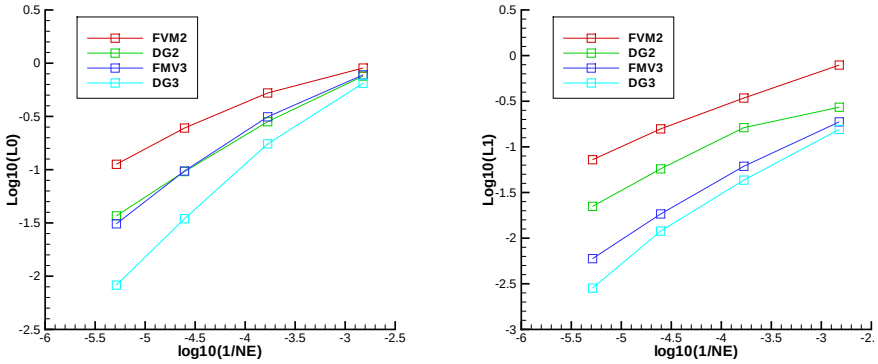
Table 4.4: Numerical Convergence Rates for FVM and DG for linear advection case at time t=1 with IC Sin(4πx) for P_1 with CFL =0.3

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	662	9.00E-1	7.87E-1			7.57E-1	2.72E-1		
10	5957	5.25E-1	3.43E-1	0.7	1.1	2.82E-1	1.62E-1	1.3	0.7
20	40398	2.46E-1	1.57E-1	1.2	1.2	9.60E-2	5.74E-2	1.7	1.6
40	193836	1.11E-1	7.23E-2	1.5	1.5	3.67E-2	2.23E-2	1.8	1.8

Table 4.5: Numerical Convergence Rates for FVM and DG for linear advection case at time t=1 with IC Sin(4πx) for P_2 with CFL =0.2

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	662	7.77E-1	1.88E-1			6.48E-1	1.55E-1		
10	5957	3.14E-1	6.13E-2	1.2	1.5	1.75E-1	4.32E-2	1.8	1.7
20	40398	9.72E-2	1.84E-2	1.8	1.9	3.46E-2	1.19E-2	2.5	2.0
40	193836	3.10E-2	5.96E-3	2.2	2.2	8.23E-3	2.83E-3	2.7	2.8

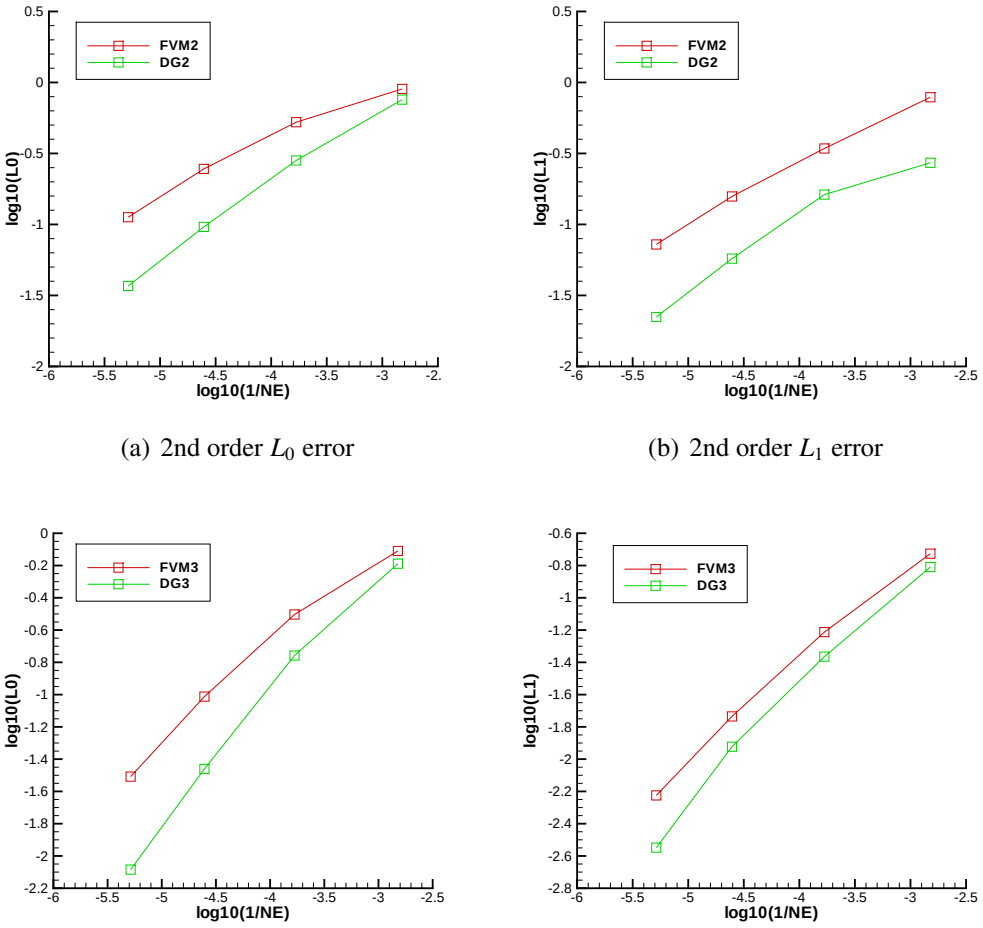
A comparative illustration is given in **Figure 4.4**, this shows the superiority of the DG3 method as opposed to the 2nd order methods.



(a) L0 error 2nd and 3rd order (b) L1 error 2nd and 3rd order

Figure 4.4: Comparison of L_0 and L_1 errors for FVM and DG, with initial conditions Sin(2πx)Sin(2πy)Sin(2πz) at T=1

Figure 4.5 illustrates the comparative convergence of the testcase for L_0 and L_1 for 2nd order and 3rd order accuracies. The linear convergence on a logarithmic becomes evident in the latter stages of the graph i.e when the mesh is refined. Once again, like in the previous testcase, the DG method displays both a smaller error and a faster convergence to the nominal order of accuracy.



(a) 2nd order L_0 error

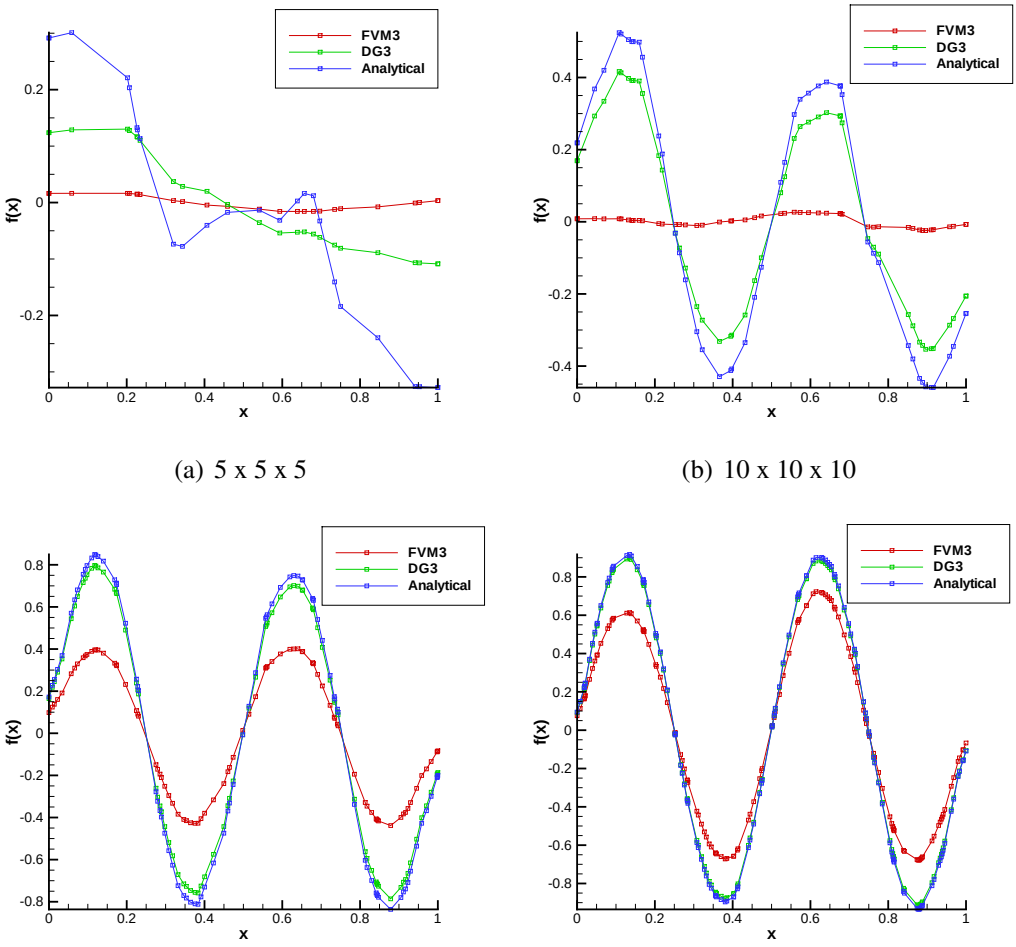
(b) 2nd order L_1 error

(c) 3rd order L_0 error

(d) 3nd order L_1 error

Figure 4.5: L_0 and L_1 errors for FVM and DG with initial conditions Sin(4πx)

Taking a cut of the $sin(4\pi x)sin(4\pi y)sin(4\pi z)$ isosurfaces at $T = 1$ at positions $z = 0.125$ and $y = 0.125$ we can extract the sine wave propagating in the x direction, which we have termed $f(x) = sin(4\pi x)$. Equally extracting $f(y) = sin(4\pi y)$ or $f(z) = sin(4\pi z)$ would have yielded the sine wave propagating in the y and z directions respectively. A graphical illustration of $f(x)$ is given in **Figure 4.6** for the four grids used in this testcase(**Table 4.1**).



(a) 5 x 5 x 5

(b) 10 x 10 x 10

(c) 20 x 20 x 20

(d) 40 x 40 x 40

Figure 4.6: Comparison of FVM3 and DG3 for profile $f(x) = Sin(4\pi x)$

Initial Condition: Sin(6πx)Sin(6πy)Sin(6πz)

Tables 4.6, 4.7 give the L_0 and L_1 errors for the advection equation which was initialised with the condition

$$U_0 = U(x, y, z, 0) = sin(6\pi x)sin(6\pi y)sin(6\pi z)$$

This was simulation was run for T=1 physical time

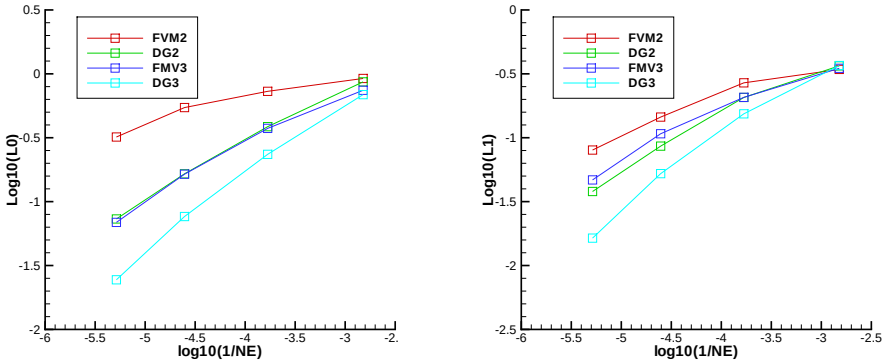
Table 4.6: Numerical Convergence Rates for FVM and DG for linear advection case at time t=1 with IC Sin(6πx) for P_1 with CFL=0.3

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	662	9.20E-1	3.43E-1			8.65E-1	3.65E-1		
10	5957	7.30E-1	2.69E-1	0.3	0.3	3.86E-1	2.07E-1	1.1	0.8
20	40398	5.44E-1	1.45E-1	0.45	1.0	1.65E-1	8.60E-2	1.3	1.4
40	193836	3.20E-1	8.01E-2	1.0	1.1	7.32E-2	1.64E-2	1.6	1.6

Table 4.7: Numerical Convergence Rates for FVM and DG for linear advection case at time t=1 with IC Sin(6πx) for P_2 with CFL=0.2

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	662	7.44E-1	3.50E-1			6.87E-1	3.67E-1		
10	5957	3.74E-1	2.07E-1	0.9	0.7	2.34E-1	1.14E-1	1.5	1.6
20	40398	1.64E-1	1.07E-1	1.3	1.0	6.65E-2	2.24E-2	2.0	2.5
40	193836	6.88E-2	4.68E-2	1.7	1.6	2.05E-2	5.32E-3	2.3	2.7

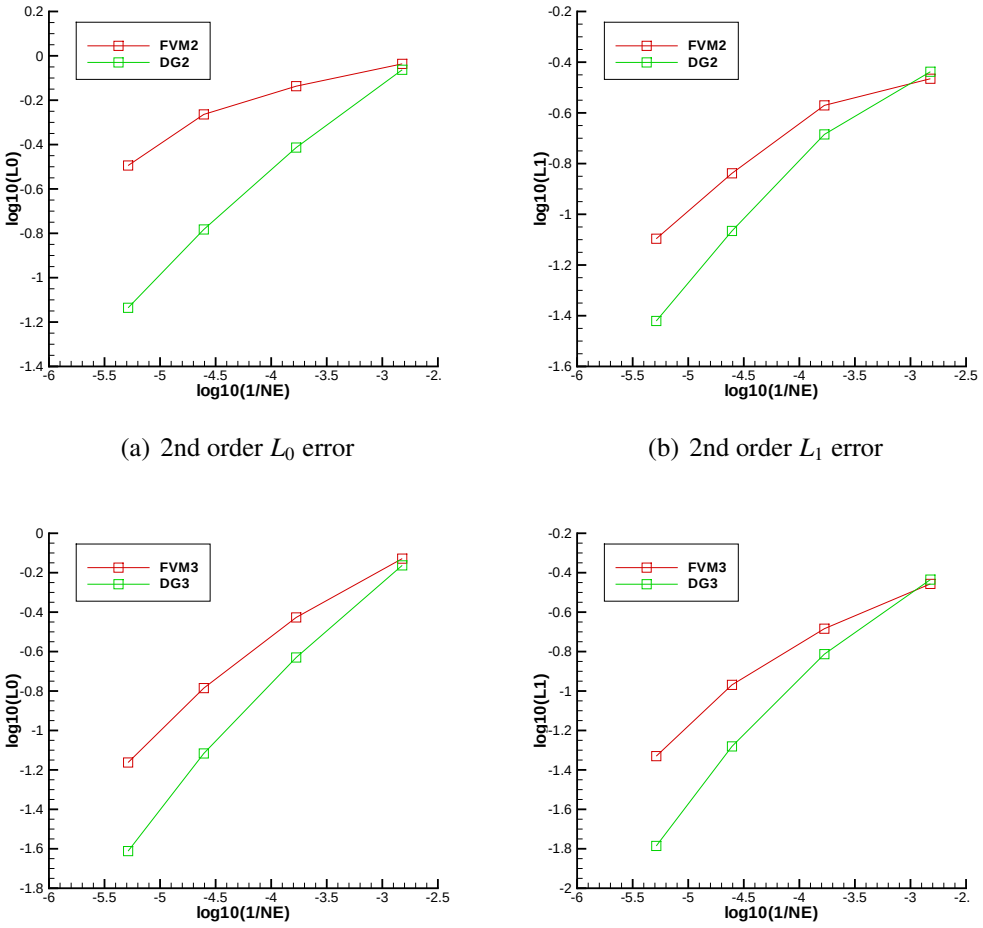
A comparative illustration is given in **Figure 4.7**, this shows the superiority of the DG3 method as opposed to the 2nd order methods.



(a) L0 error 2nd and 3rd order (b) L1 error 2nd and 3rd order

Figure 4.7: Comparison of L_0 and L_1 errors for FVM and DG, with initial conditions Sin(2πx)Sin(2πy)Sin(2πz) at T=1

Figure 4.8 illustrates the comparative convergence of the testcase for L_0 and L_1 for 2nd order and 3rd order accuracies. Similar to the previous testcases for the linear advection equation, the linear convergence on a logarithmic becomes evident. Furthermore, the DG method displays both a smaller error and a faster convergence to the nominal order of accuracy than the FVM method.



(a) 2nd order L_0 error

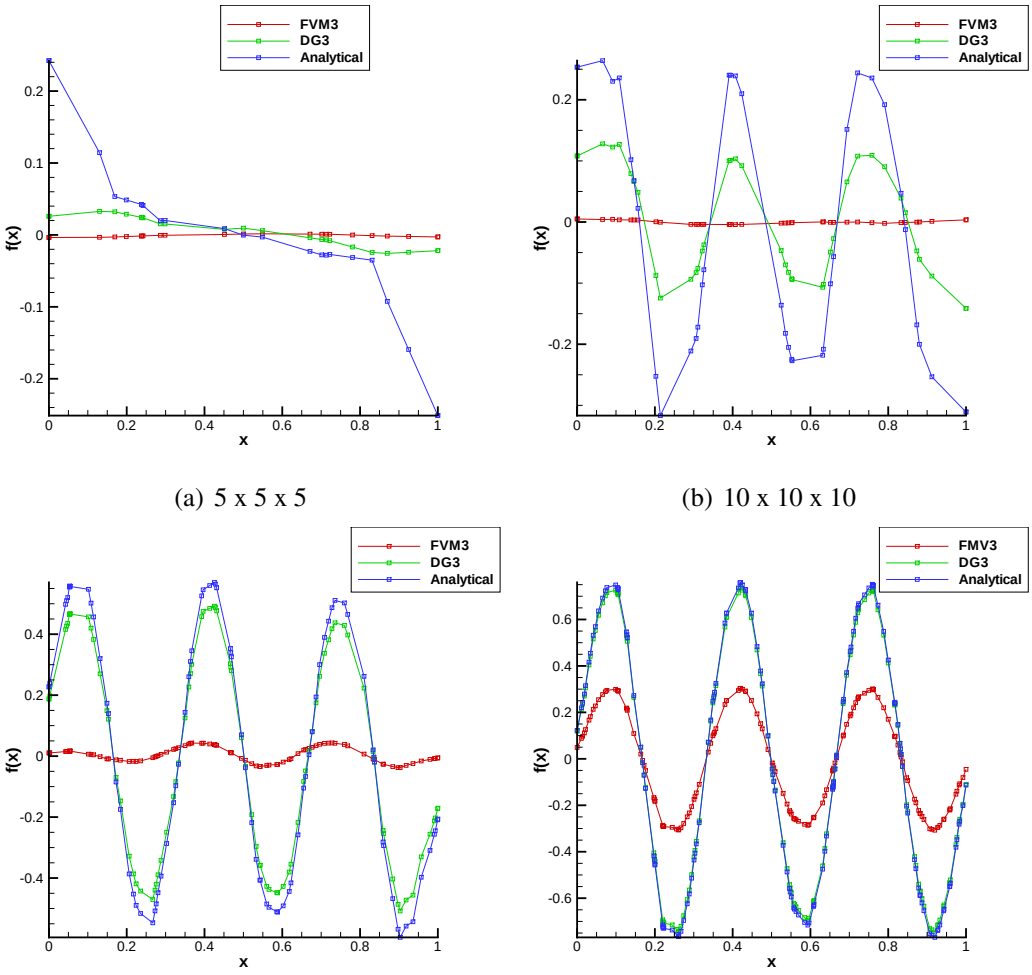
(b) 2nd order L_1 error

(c) 3rd order L_0 error

(d) 3nd order L_1 error

Figure 4.8: L_0 and L_1 errors for FVM and DG with initial conditions Sin(6πx)

Taking a cut of the $sin(6\pi x)sin(6\pi y)sin(6\pi z)$ isosurfaces at $T = 1$ at positions $z = 0.0625$ and $y = 0.0625$ we can extract the sine wave propagating in the x direction, which we have termed $f(x) = sin(6\pi x)$. Equally extracting $f(y) = sin(6\pi y)$ or $f(z) = sin(6\pi z)$ would have yielded the sine wave propagating in the y and z directions respectively. A graphical illustration of $f(x)$ is given in **Figure 4.9** for the four grids used in this testcase(**Table 4.1**). It is apparent from these graphs that the FVM has relatively poor resolution when compared to the DG method.



(a) 5 x 5 x 5

(b) 10 x 10 x 10

(c) 20 x 20 x 20

(d) 40 x 40 x 40

Figure 4.9: Comparison of FVM3 and DG3 for profile $f(x) = Sin(6\pi x)$

Computational Time

The time taken for the above simulations is given below, an average from the test-cases for the three initial conditions was taken for physical time T=1 for the four grids(**Table 4.1**).

Table 4.8: Computational time for linear advection equation for FVM2 and DG2

N_G	FVM2	DG2
5 x 5 x 5	7sec	26sec
10 x 10 x 10	89sec	452sec
20 x 20 x 20	139sec	3549sec
40 x 40 x 40	14277sec	33279sec

Table 4.9: Computational time for linear advection equation for FVM3 and DG3

N_G	FVM3	DG3
5 x 5 x 5	34sec	307sec
10 x 10 x 10	711sec	5500sec
20 x 20 x 20	11502sec	85019sec
40 x 40 x 40	105242sec	789315sec

FVM clearly requires less computational time than the DG method, for the 2nd order methods (**Table 4.8**) FVM is approximately 2 times faster than DG and for 3rd order methods (**Table 4.9**) FVM is approximately 7 times faster than the DG method.

4.1.2 Euler Equations

The Euler Equations[1]

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial F(\mathbf{U})}{\partial x} + \frac{\partial G(\mathbf{U})}{\partial y} + \frac{\partial H(\mathbf{U})}{\partial z} = 0$$

With the initial conditions

$$(\rho, u, v, w, p)(x, y, z, 0) = (0.2 sin\left(\pi \frac{x + y + z}{3}\right), 1, 1, 1, 1)$$

Are solved on four grids (**Table 4.10**) for 2nd and 3rd order accuracy. The above problem has an analytical solution which makes it possible to compare against the numerical solution and establish the L_0 and L_1 errors, thus the nominal order of accuracy. The domain size is $[-3, 3]^3$ and periodic boundary conditions are enforced in all directions.

Table 4.10: Grid information for Euler equation testcase

	N_G	N_E	2^{nd} order: N_{DOF}	3^{rd} order: N_{DOF}
Grid 1	5 x 5 x 5	1908	7632	19080
Grid 2	10 x 10 x 10	9924	39696	99240
Grid 3	20 x 20 x 20	70103	280412	701030
Grid 4	40 x 40 x 40	472971	1891884	4729710

Table 4.10, also shows the total number of degrees of freedom (N_{DOF}) for the 2nd and 3rd order DG method, along with the Number of grid points (N_G) and number of elements (N_E). For 1st order DG, N_{DOF} is the same as the number of elements. And for FVM, the N_{DOF} does not change as the order of accuracy increases - FVM has one degree of freedom per element per variable.

The order of the temporal accuracy and CFL number with respect to the spatial accuracy is set according to the findings of[12],these have been replicated and are available in the appendix. For 2nd order accuracy, we use a 2nd order Runge-Kutta method, for 3rd order a 3rd order Runge-kutta method, the CFL number was approximated using the condition

$$CFL \approx \frac{1}{2l + 1}$$

Where l is the order of the polynomial.

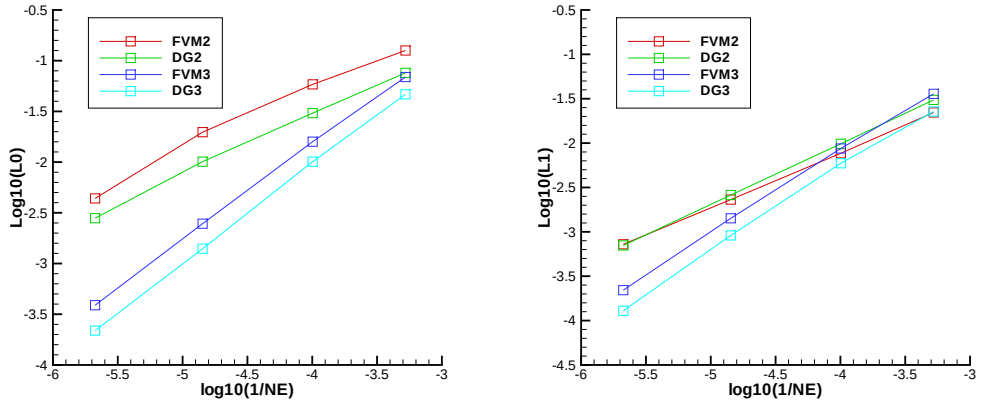
[1] Please consult the appendix for definition of F ,G ,H and **U**

Table 4.11: Numerical Convergence Rates for FVM and DG for the Euler case for ρ at T=1 for P_1 with CFL=0.3

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	1,908	1.26E-01	2.21E-02			7.59E-02	3.07E-02		
10	9,924	5.84E-02	7.68E-03	1.4	1.9	3.04E-02	9.83E-03	1.7	2.3
20	70,103	1.97E-02	2.31E-03	1.7	1.8	1.01E-02	2.60E-03	1.7	1.9
40	472,971	4.38E-03	7.24E-04	2.3	1.8	2.80E-03	7.01E-04	2.0	2.0

Table 4.12: Numerical Convergence Rates for FVM and DG for the Euler case for ρ at T=1 for P_2 with CFL=0.2

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	1908	6.90E-02	3.59E-02			4.66E-02	2.27E-02		
10	9924	1.59E-02	8.69E-03	2.7	2.5	1.01E-02	5.92E-03	2.8	2.4
20	70103	2.47E-03	1.42E-03	2.9	2.8	1.40E-03	9.10E-04	3.0	2.9
40	472971	3.88E-04	2.20E-04	2.9	2.9	2.18E-04	1.29E-04	2.9	3.1



(a) L0 error 2nd and 3rd order

(b) L1 error 2nd and 3rd order

Figure 4.10: Comparison of L_0 and L_1 errors for FVM and DG, for Euler Equations with smooth initial conditions at T=1

The L_0 and L_1 error (**Table 4.11** and **Table 4.12**) for the P_1 and P_2 approximations show a convergence to the nominal order of accuracy. Plotting this L_0 and L_1 error on a logarithmic scale, shows the linear convergence to the nominal order of accuracy for increasing number of elements (**Figure 4.10**).

4.2 Barth-Jespersen Limiter

The smooth problems mentioned in the previous section did not require a limiter, because as illustrated by the profiles of the sinewaves of different wavenumbers there was no Gibb's oscillations that were present. Furthermore, the FVM and DG methods for 2nd and 3rd order accuracies converged to the nominal order of accuracy for the $sin(2\pi x)sin(2\pi y)sin(2\pi z)$ initial condition, and the DG3 method showed convergence even in the more demanding testcases with initial conditions $sin(4\pi x)sin(4\pi y)sin(4\pi z)$, $sin(6\pi x)sin(6\pi y)sin(6\pi z)$. In this section we will apply the Barth-Jespersen limiter to the three testcases we investigated in the previous section. We know from previous investigations that a TVD limiter like the Barth-Jespersen will degrade the resolution convergence of the solutions to the nominal order of accuracy in the smooth regions. Our aim in this section is to compare the effect of the Barth-Jespersen limiter on the FVM2 and DG2.

Initial Condition: Sin(2πx)Sin(2πy)Sin(2πz)

Table 4.13: Numerical Convergence Rates for TVD-FVM and TVD-DG for linear advection case at time T=1 with IC Sin(2πx)Sin(2πy)Sin(2πz) for P_1 with CFL=0.3

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	662	9.98E-1	7.45E-1			9.54E-1	6.95E-1		
10	5957	7.87E-1	6.28E-1	0.32	0.233	7.67E-1	5.63E-1	0.30	0.29
20	40398	5.79E-1	4.57E-1	0.48	0.50	5.43E-1	4.19E-1	0.54	0.46
40	193836	3.27E-1	2.56E-1	1.09	1.11	3.02E-1	2.23E-1	1.12	1.20

Table 4.13 shows the convergence of the FVM2-TVD and DG2-TVD, as is evident when compared to **Table 4.2** for FVM2 and DG2 (without the use of a Barth-Jespersen limiter) the error is greater with the use of the limiter. The convergence to the nominal order of accuracy has also been degraded. The convergence and error of the FVM and DG method is similar when a Barth-Jespersen limiter is applied (**Table 4.12**). Similar to the unlimited case we take a cut of the isosurfaces of $sin(2\pi x)sin(2\pi y)sin(2\pi z)$ at points $z = 0.25$ and $y = 0.25$ to extract the sine wave propogating in the x-direction which is termed $f(x) = sin2\pi x$. **Table 4.2** shows the profile of the FVM2-TVD and DG2-TVD sine waves propogating in the x-direction at T=1, and compare this to the 1st order approximation and analytical solution. Even with the Barth-Jespersen limiter degrading the resolution of the 2nd order FVM and DG approximation, they are still produce a better resolution than the 1st order approximation.

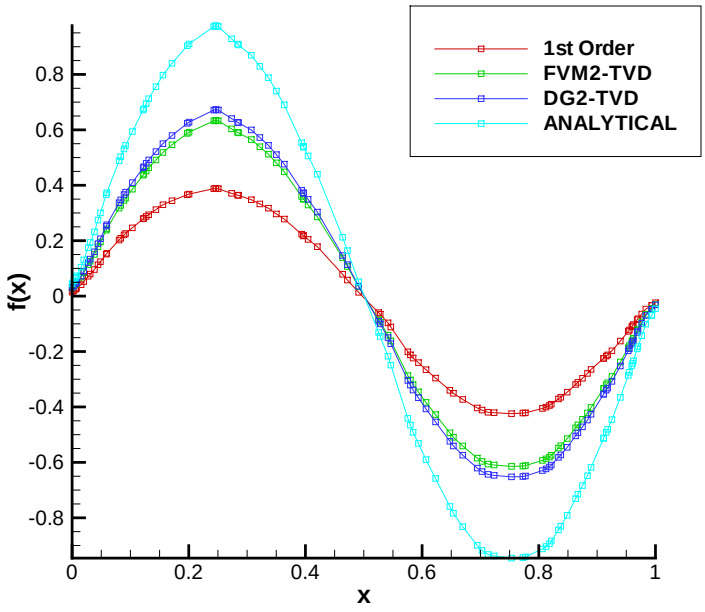
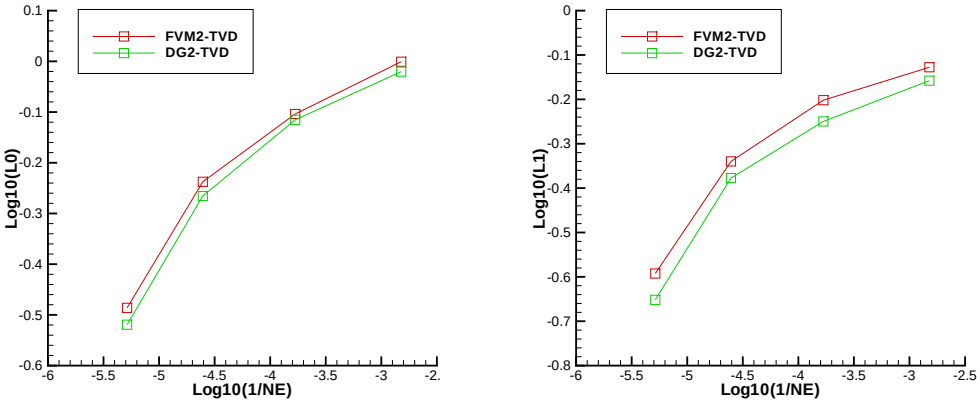


Figure 4.11: Comparison of FVM2-TVD and DG2-TVD for profile $f(x) = Sin(2\pi x)$



(a) L0 error (b) L1 error

Figure 4.12: 2nd order Comparison of L_0 and L_1 errors for FVM-TVD and DG-TVD, with smooth initial conditions Sin(2πx)Sin(2πy)Sin(2πz) at T=1

Initial Condition: Sin(4πx)Sin(4πy)Sin(4πz)

Table 4.14: Numerical Convergence Rates for TVD-FVM and TVD-DG for linear advection case at time T=1 with IC Sin(4πx)Sin(4πy)Sin(4πz) for P_1 with CFL=0.3

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	662	1.02	8.72E-1			9.84E-1	7.72E-1		
10	5957	8.49E-1	7.04E-1	0.25	0.29	7.83E-1	6.35E-1	0.31	0.27
20	40398	6.49E-1	5.23E-1	0.42	0.47	5.78E-1	4.62E-1	0.48	0.50
40	193836	4.26E-1	3.30E-1	0.80	0.88	3.42E-1	2.57E-1	1.00	1.12

Table 4.13 shows the convergence of the FVM2-TVD and DG2-TVD, as is evident when compared to **Table 4.2** for FVM2 and DG2 (without the use of a Barth-Jespersen limiter) the error is greater with the use of the limiter. The convergence to the nominal order of accuracy has also been degraded. The convergence and error of the FVM and DG method is similar when a Barth-Jespersen limiter is applied (**Table 4.12**). Similar to the unlimited case we take a cut of the isosurfaces of $sin(2\pi x)sin(2\pi y)sin(2\pi z)$ at points $z = 0.25$ and $y = 0.25$ to extract the sine wave propogating in the x-direction which is termed $f(x) = sin2\pi x$. **Table 4.2** shows the profile of the FVM2-TVD and DG2-TVD sine waves propogating in the x-direction at T=1, and compare this to the 1st order approximation and analytical solution. Even with the Barth-Jespersen limiter degrading the resolution of the 2nd order FVM and DG approximation, they are still produce a better resolution than the 1st order approximation.

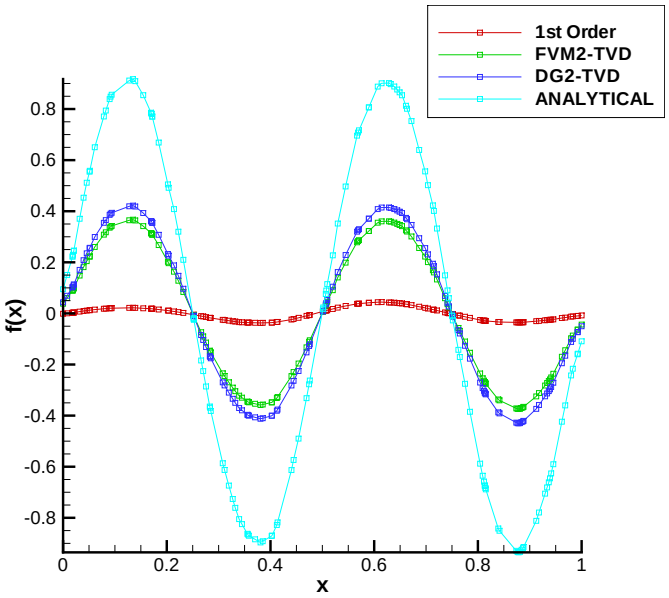
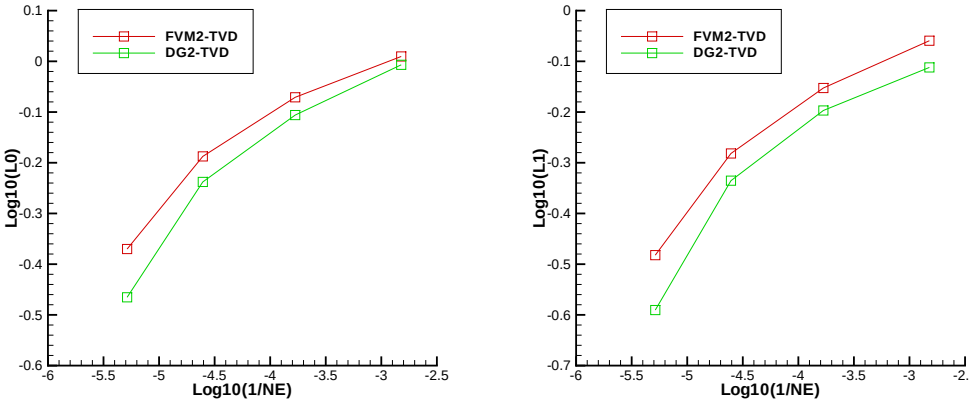


Figure 4.13: Comparison of FVM2-TVD and DG2-TVD for profile $f(x) = Sin(4\pi x)$



(a) L0 error (b) L1 error

Figure 4.14: 2nd order Comparison of L_0 and L_1 errors for FVM-TVD and DG-TVD, with smooth initial condition Sin(4πx)Sin(4πy)Sin(4πz) at T=1

Initial Condition: Sin(6πx)Sin(6πy)Sin(6πz)

Table 4.15: Numerical Convergence Rates for TVD-FVM and TVD-DG for linear advection case at time T=1 with IC Sin(6πx)Sin(6πy)Sin(6πz) for P_1 with CFL=0.3

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	662	1.09	8.93E-1			1.01	8.75E-1		
10	5957	8.98E-1	7.24E-1	0.27	0.29	8.36E-1	6.93E-2	0.26	0.32
20	40398	7.13E-1	5.87E-1	0.36	0.33	6.47E-1	5.21E-2	0.40	0.45
40	193836	5.23E-1	4.15E-1	0.59	0.66	3.89E-1	2.93E-2	0.92	1.10

Table 4.13 shows the convergence of the FVM2-TVD and DG2-TVD, as is evident when compared to **Table 4.2** for FVM2 and DG2 (without the use of a Barth-Jespersen limiter) the error is greater with the use of the limiter. The convergence to the nominal order of accuracy has also been degraded. The convergence and error of the FVM and DG method is similar when a Barth-Jespersen limiter is applied (**Table 4.12**). Similar to the unlimited case we take a cut of the isosurfaces of $sin(2\pi x)sin(2\pi y)sin(2\pi z)$ at points $z = 0.25$ and $y = 0.25$ to extract the sine wave propogating in the x-direction which is termed $f(x) = sin2\pi x$. **Table 4.2** shows the profile of the FVM2-TVD and DG2-TVD sine waves propogating in the x-direction at T=1, and compare this to the 1st order approximation and analytical solution. Even with the Barth-Jespersen limiter degrading the resolution of the 2nd order FVM and DG approximation, they are still produce a better resolution than the 1st order approximation.

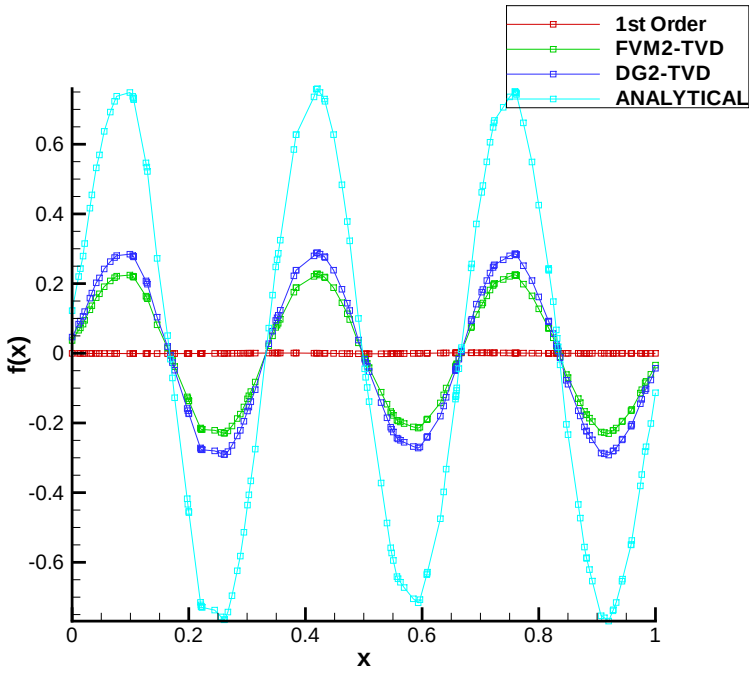
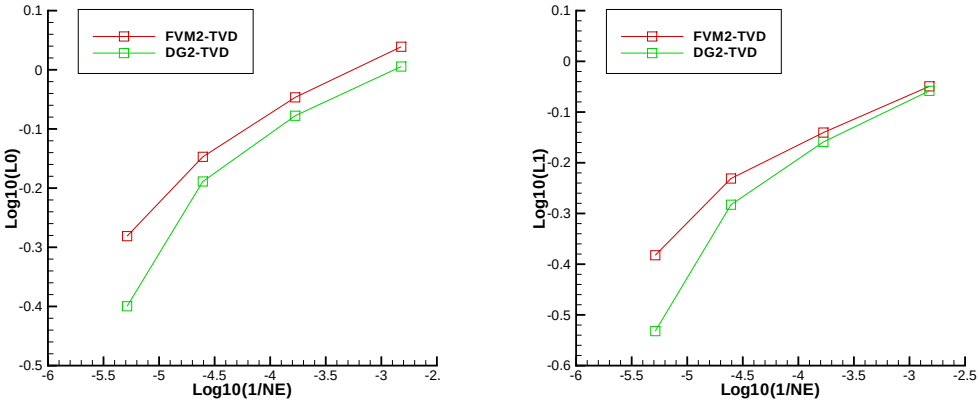


Figure 4.15: Comparison of FVM2-TVD and DG2-TVD for profile $f(x) = Sin(6\pi x)$



(a) L0 error (b) L1 error

Figure 4.16: 2nd order Comparison of L_0 and L_1 errors for FVM-TVD and DG-TVD, with smooth initial condition Sin(6πx)Sin(6πy)Sin(6πz) at T=1

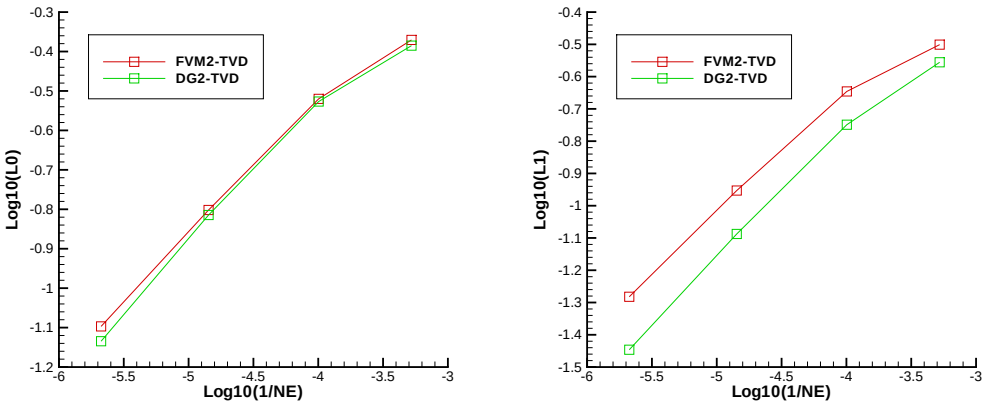
Euler Equations

Similar to the Analysis for the Euler Equation conducted in the previous chapter we solve the Euler equations on four grids (**Table 4.10**) for 2nd and order accuracy with a Barth-Jespersen limiter for the finite volume method and Discontinuous Galerkin method, FVM2-TVD and DG2-TVD respectively. The above problem has an analytical solution which makes it possible to compare against the numerical solution and establish the L_0 and L_1 errors, thus the nominal order of accuracy. The domain size is $[-3,3]^3$ and periodic boundary conditions are enforced in all directions. The order of the temporal accuracy and CFL number with respect to the spatial accuracy is set at 2nd order Runge-Kutta method with CFL=0.3

Table 4.16: Numerical Convergence Rates for FVM and DG for the Euler case for ρ at T=1 for P_1 with CFL=0.3

		FVM				DG			
N_G	Tetrahedrons	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
5	1908	4.26E-01	3.16E-01			4.12E-01	2.78E-01		
10	9924	3.02E-01	2.26E-01	0.6	0.6	2.97E-01	1.78E-01	0.6	0.8
20	70103	1.58E-01	1.11E-01	1.0	1.1	1.53E-01	8.17E-02	1.0	1.2
40	472971	8.00E-02	5.22E-02	1.1	1.2	7.34E-02	3.58E-02	1.2	1.3

Table 4.16 shows the error and nominal order of accuracy for the L_0 and L_1 errors for FVM2-TVD and DG2-TVD, this shows the denigration of convergence to the nominal order of accuracy and increase in error when compared to **Table 4.11** when a Barth-Jespersen limiter is applied to a smooth solution.



(a) L0 error (b) L1 error

Figure 4.17: L_0 and L_1 errors for FVM-TVD and DG-TVD, for Euler Equations with smooth initial conditions at T=1 (RK=2 CFL=0.3)

Table 4.17 shows the near comparison for the L_0 error, and the non-linear nature of the convergence.

4.3 Discontinuous Testcases

4.3.1 Shock tubecase

The Sod shock tube case was proposed by Gary Sod[62]. This can also be viewed as a Riemann problem, as there is a discontinuity separating different states either side of the discontinuity. A diaphragm separating the gases which in this particular case is located at the centre of the domain ($x = 0$) is broken allowing the movement of the gases. Initially the gases either side of the diaphragm are at rest, therefore $u = v = w = 0$. However, the pressures and densities of the gases are different, therefore, the flow movement is initiated using a pressure gradient. Strictly speaking this shock tube problem is 1D, however, it forms an important first port-of-call for testing a solver's ability to successfully capture shock waves. Outflow boundary conditions were applied at $x = 0$ and $x = 1$. As there is no variation in the y and z directions of the flow we are free to apply any boundary condition we want at the y and z boundaries - in this particular case we applied wall conditions, although applying periodic conditions could have been applied. A TVD Barth-Jespersen limiter was applied to the linear forms of the 2nd order FVM and DG methods. A 2nd order Runge-Kutta Time-stepping was employed with $CFL = 0.3$. This shocktube problem was solved on four grids (**Table 4.17**) to conduct a grid convergence study.

Table 4.17: Grid information for Sod's shocktube

Number of gridpoints	elements
$N_G = 10$	10645 tetrahedrons
$N_G = 15$	22607 tetrahedrons
$N_G = 25$	124184 tetrahedrons
$N_G = 50$	464274 tetrahedrons

The wave structure of the Sod shocktube problem comprises of a left travelling rarefaction, a contact discontinuity travelling to the right and a right travelling shock wave as illustrated in **Figure 4.18**.

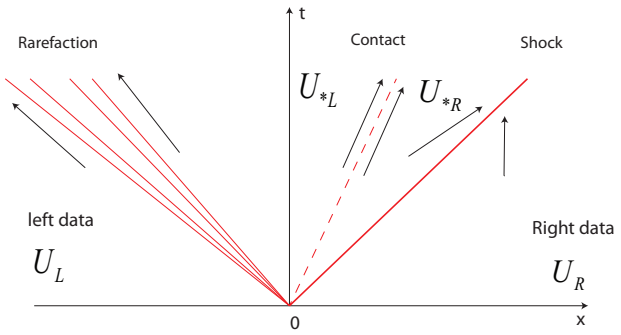
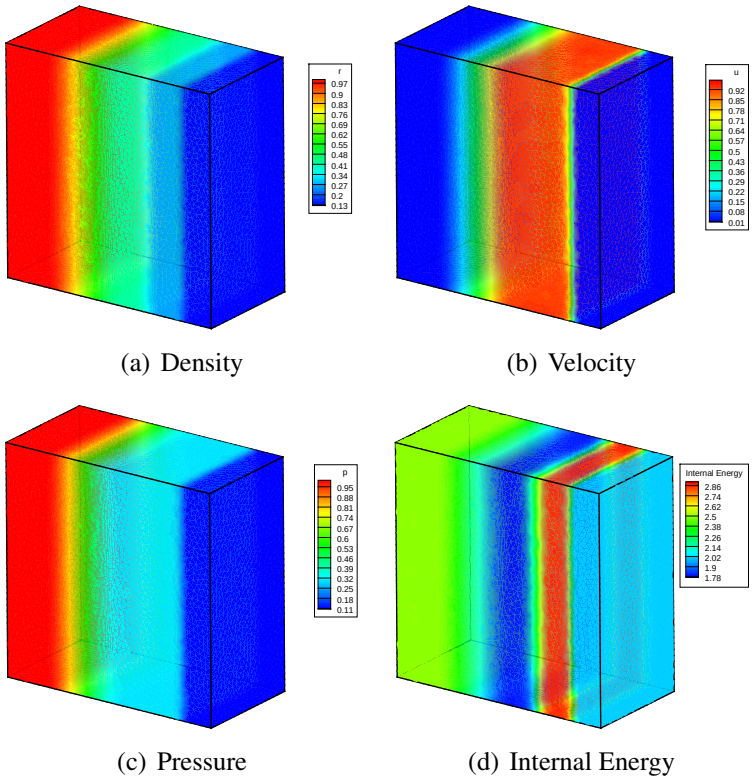


Figure 4.18: Wave structure for the Sod shock tube case

The grid used in the simulation has the dimension $[0,1]_x \times [0,1]_y \times [0,0.5]_z$, and the initial conditions are

$$(\rho, u, v, w, p)(x, y, z, 0) = \begin{cases} (1, 0, 0, 0, 1) & \text{if } x \leq 0 \\ (0.125, 0, 0, 0, 0.1) & \text{if } x > 0 \end{cases}$$

After T=0.2 for the 50x50x50 grid, the Sod's shocktube case produces the isocontours **Figure 4.19**.

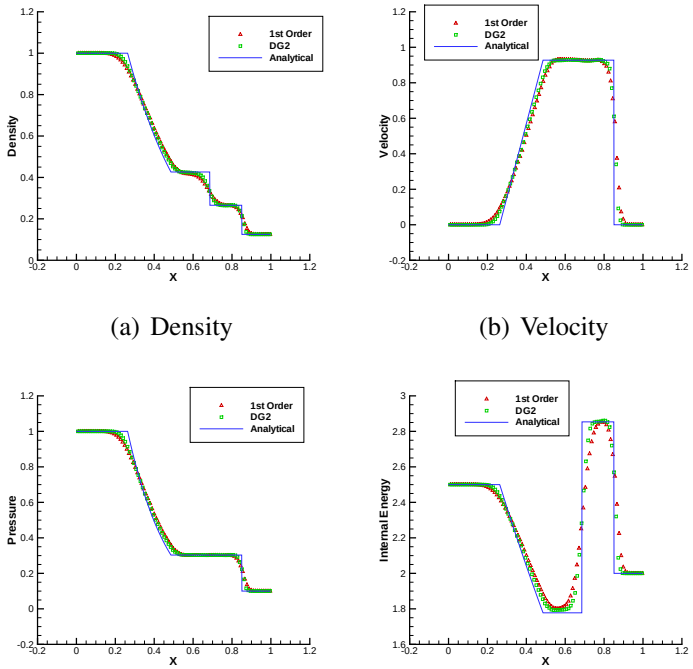


(a) Density (b) Velocity

(c) Pressure (d) Internal Energy

Figure 4.19: DG2-TVD isocontours of Sod shocktube case at T=0.2

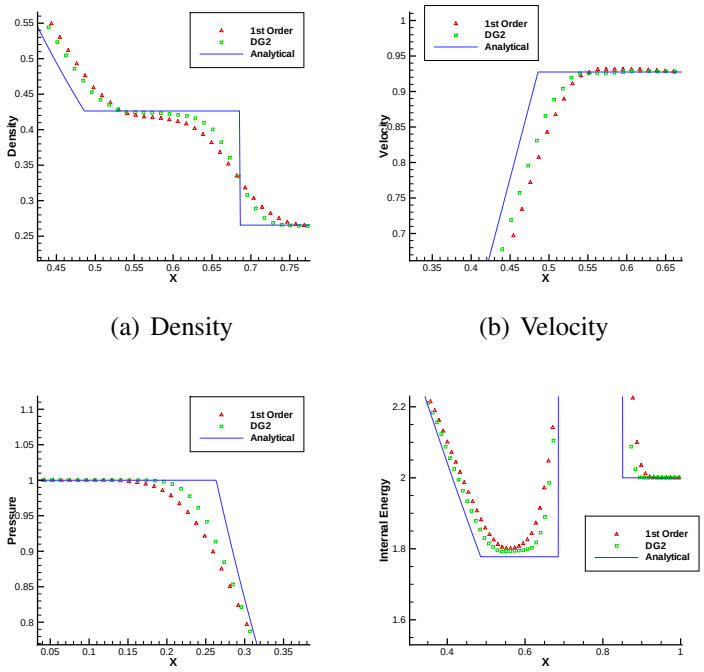
Firstly a comparison between a 1st order method and the DG2-TVD method was conducted. Extracting the radial solutions from the isocontours at the plane of $x = 0.5, y = 0.5$. The results for DG2-TVD were compared to the 1st order solution Figure 4.20, Figure 4.21, which shows an improvement over the 1st order results, but not considerably, this is in line with theorem that TVD methods in mutlidimensions are at most 1st order accurate, hence the poor resolution.



(a) Density (b) Velocity

(c) Pressure (d) Internal Energy

Figure 4.20: Comparison of 1st order and DG2 accuracy for Sod shock tube case at t=0.2

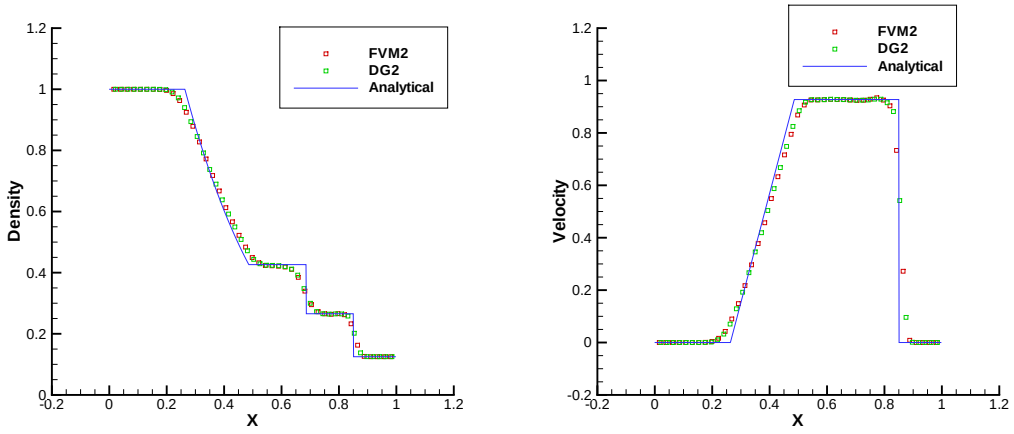


(a) Density (b) Velocity

(c) Pressure (d) Internal Energy

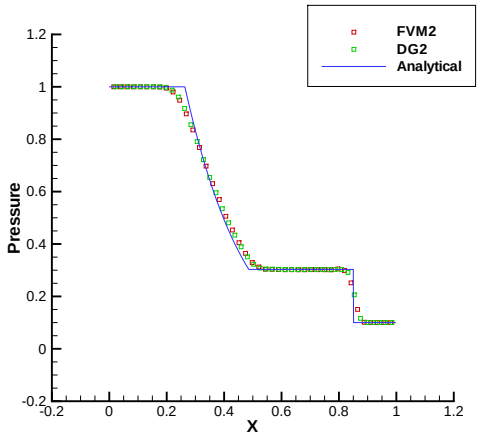
Figure 4.21: Zoomed in 1st order and DG2 accuracy for Sod shock tube case at t=0.2

The convergence study was conducted in 1D, the data for which was extracted from the contours(4.20) at $x = 0.5$ and $y = 0.5$, the resulting radial comparison between the FVM2-TVD and DG2-TVD and the analytical solution is illustrated in **Figure 4.22**.

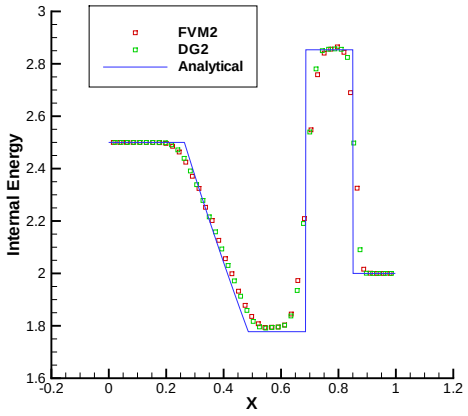


(a) Density

(b) Velocity



(c) Pressure



(d) Internal Energy

Figure 4.22: Comparison of FVM and DG for Sod shock tube case

100 data points were extracted from the contours, for both FVM and DG, and the results were compared to the analytical solution to establish the nominal order of accuracy. The resulting tables (4.18, 4.19, 4.20, 4.21) show the errors for both FVM and DG for the four grids, and the resulting nominal order of accuracy.

Table 4.18: Numerical Convergence Rates of FVM and DG for ρ of Sod's shock tube case

	FVM				DG			
N_G	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
10	1.15E-01	3.40E-02	-	-	1.13E-01	3.54E-02	-	-
15	1.10E-01	3.02E-02	8.67E-02	2.32E-01	1.10E-01	2.95E-02	5.24E-02	3.56E-01
25	1.07E-01	2.44E-02	5.12E-02	3.84E-01	1.06E-01	2.38E-02	5.60E-02	3.85E-01
50	1.03E-01	1.58E-02	7.65E-02	9.71E-01	1.02E-01	1.54E-02	8.20E-02	9.75E-01

Table 4.19: Numerical Convergence Rates of FVM and DG for u of Sod's shock tube case

	FVM				DG			
N_G	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
10	5.25E-01	6.78E-02	-	-	5.23E-01	6.54E-02	-	-
15	4.55E-01	5.77E-02	2.79E-01	3.15E-01	4.53E-01	5.51E-02	2.80E-01	3.34E-01
25	3.87E-01	4.54E-02	2.90E-01	4.28E-01	3.82E-01	4.10E-02	3.05E-01	5.29E-01
50	3.11E-01	2.73E-02	4.91E-01	1.14	3.06E-01	2.40E-02	4.98E-01	1.20

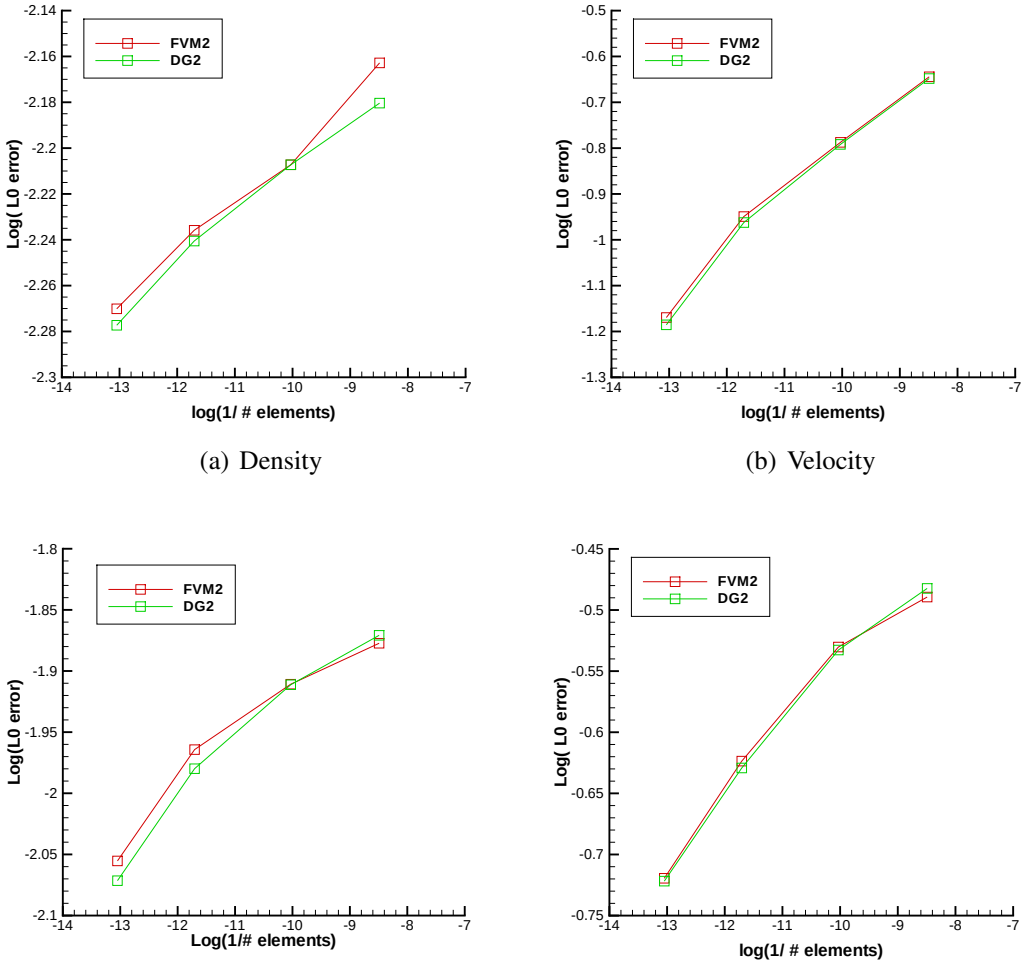
Table 4.20: Numerical Convergence Rates of FVM and DG for p of Sod's shock tube case

	FVM				DG			
N_G	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
10	1.53E-01	3.47E-02	-	-	1.54E-01	3.39E-02	-	-
15	1.49E-01	2.91E-02	5.21E-02	3.41E-01	1.50E-01	2.88E-02	5.00E-02	3.21E-01
25	1.43E-01	2.39E-02	7.62E-02	3.54E-01	1.42E-01	2.24E-02	1.01E-01	4.48E-01
50	1.28E-01	1.45E-02	2.42E-01	1.12	1.26E-01	1.39E-02	2.64E-01	1.07

Table 4.21: Numerical Convergence Rates of FVM and DG for IE of Sod's shock tube case

	FVM				DG			
N_G	L_0	L_1	O_{L_0}	O_{L_1}	L_0	L_1	O_{L_0}	O_{L_1}
10	6.13E-01	1.82E-01	-	-	6.17E-01	1.77E-01	-	-
15	5.88E-01	1.68E-01	7.97E-02	1.56E-01	5.87E-01	1.65E-01	9.82E-02	1.37E-01
25	5.36E-01	1.39E-01	1.67E-01	3.39E-01	5.33E-01	1.35E-01	1.73E-01	3.59E-01
50	4.87E-01	9.98E-02	2.14E-01	7.40E-01	4.86E-01	9.94E-02	2.07E-01	6.84E-01

The L_0 and L_1 errors can be illustrated on a logarithm scale in figures (4.23, 4.24), where the gradient represents the nominal order of accuracy.



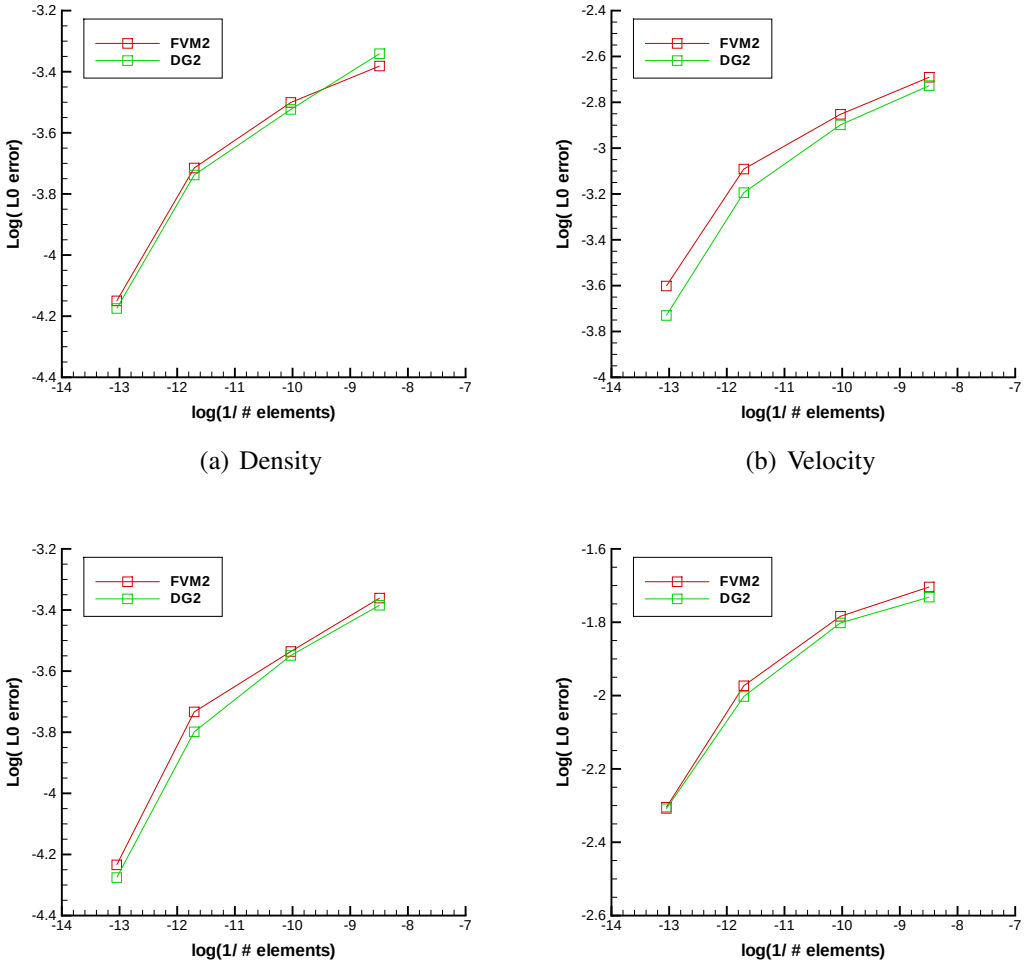
(a) Density

(b) Velocity

(c) Pressure

(d) Internal Energy

Figure 4.23: Comparison of L0 error for FVM and DG for Sod's shocktube case



(a) Density

(b) Velocity

(c) Pressure

(d) Internal Energy

Figure 4.24: Comparison of L1 error for FVM and DG for Sod's shocktube case

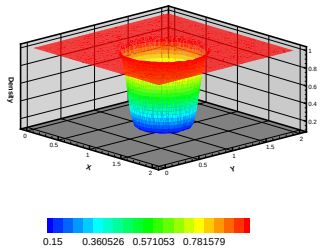
4.4 Implosion

The implosion test case is conducted on a grid of size $[0, 2]^3$, which is discretised with a total 455675 tetrahedrons. Outflow conditions are imposed in all directions. The initial conditions consist of two regions: a low density sphere of radius 0.4 surrounded by a higher density region - initiating an implosion. The Euler equations are solved with the following initial conditions

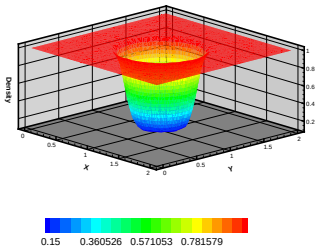
$$(\rho, u, v, w, p)(x, y, z, 0) = \begin{cases} (1, 0, 0, 0, 1) & \text{if } r > 0.4 \\ (0.125, 0, 0, 0, 0.1) & \text{if } r \leq 0.4 \end{cases}$$

where $r^2 = x^2 + y^2 + z^2$

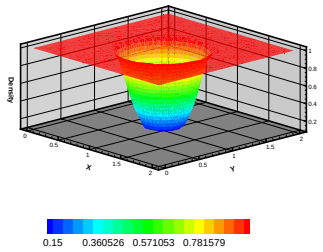
The implosion process arises in a variety of important physical problems, including thermonuclear weapons and type II supernovas. However, this implosion test case is used to demonstrate the robustness of the solver rather than a detailed analysis. Unlike the analogous explosion test case, where discontinuities are moving away, in the implosion test case the discontinuities are moving towards each other. As a result the discontinuities collide with one another, until an explosion occurs. A 2nd order DG method with 2nd order Runge-Kutta is used with CFL=0.3, and a Barth-Jespersen TVD limiter is used to suppress the Gibb's oscillations. The purpose of this testcase is not to conduct a comparison with the FVM2-TVD method as the analytical solution for implosion does not exist for 1D radial solution. Moreover this study is intended to test the robustness of the DG solver; it is a form of verification. The development of the density profile in 3D upto T=0.4 for the implosion is given in **Figure 4.25**, the same profile from a 2D prospective is given in **Figure 4.26**.



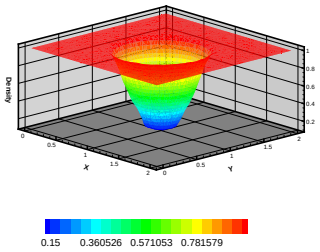
(a) t=0.0225



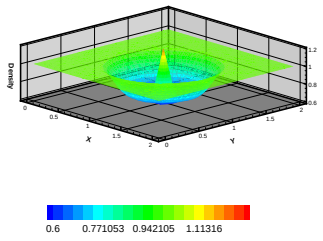
(b) t=0.0451



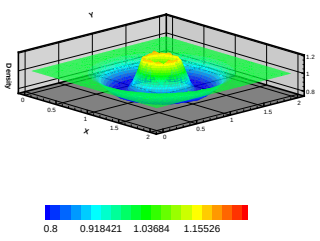
(c) t=0.0676



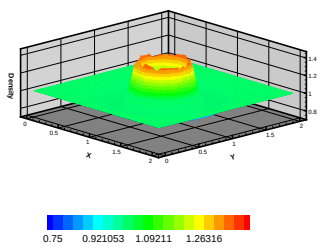
(d) t=0.0899



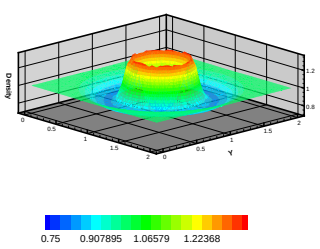
(e) t=0.179



(f) t=0.270

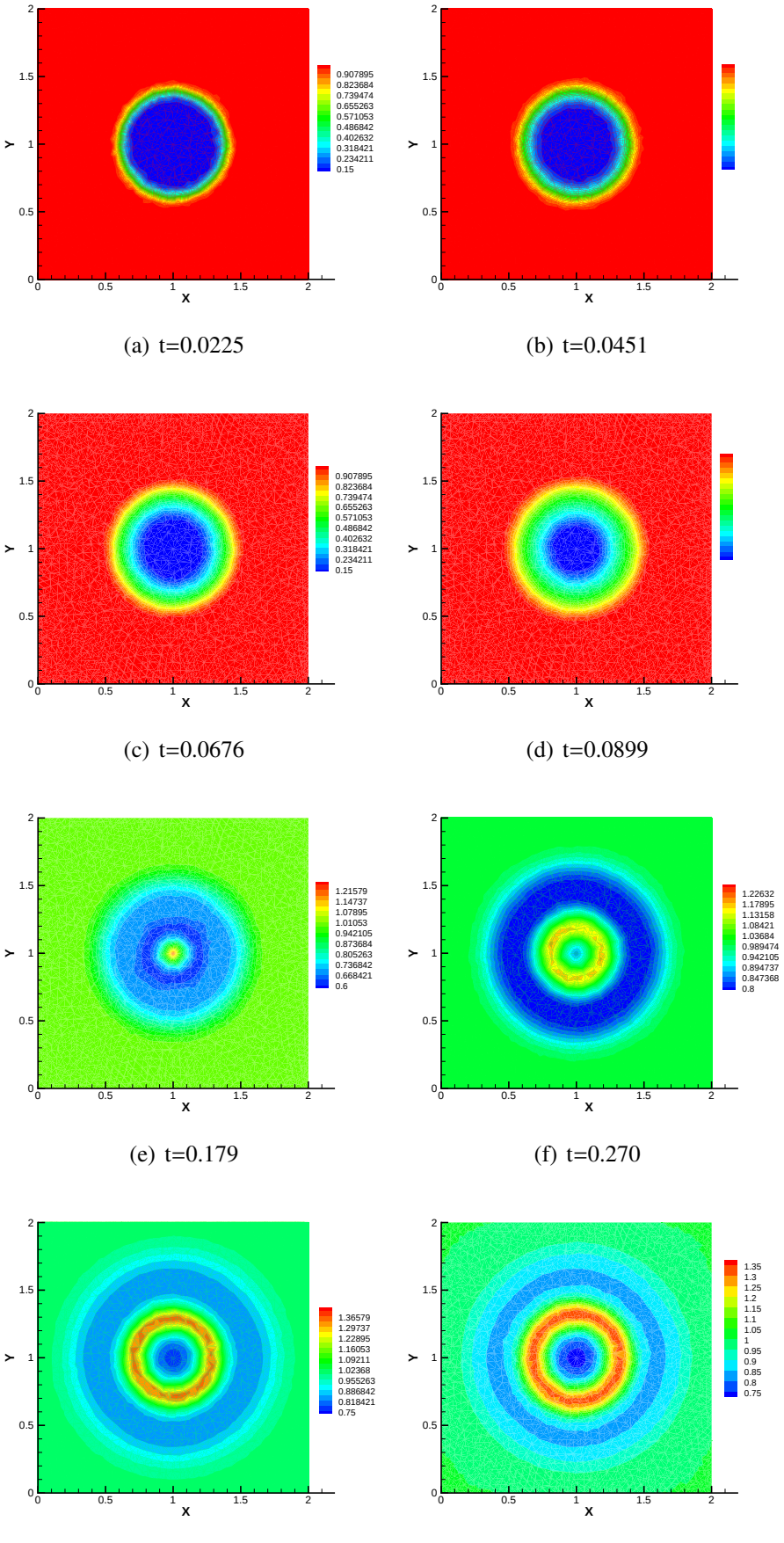


(g) t=0.36



(h) t=0.4

Figure 4.25: Density profile taken at z=1 for implosion testcase for DG2-TVD upto T=0.4



(a) t=0.0225 (b) t=0.0451

(c) t=0.0676 (d) t=0.0899

(e) t=0.179 (f) t=0.270

(g) t=0.36 (h) t=0.4

Figure 4.26: 2D cut of density taken at z=1 for implosion testcase for DG2-TVD upto T=0.4

4.5 Explosion

The Explosion test case was conducted on a grid of $[0, 2]^3$, Outflow conditions were imposed in all directions. The initial conditions consist of two regions: a high density sphere of radius 0.4 surrounded by a lower density region - initiating an explosion. The Euler equations are solved with the following initial conditions

$$(\rho, u, v, w, p)(x, y, z, 0) = \begin{cases} (1, 0, 0, 0, 1) & \text{if } r \leq 0.4 \\ (0.125, 0, 0, 0, 0.1) & \text{if } r > 0.4 \end{cases}$$

where $r^2 = x^2 + y^2 + z^2$

A 2^{nd} order FVM and DG method with 2nd order Runge-Kutta, CFL=0.2 are used, and a Barth-Jespersen TVD limiter is used to suppress the Gibb's oscillations. The solution to the problem consists of the spherical shock wave and contact wave propagating away from the centre of the explosion and conversely a smooth rarefaction travelling towards the centre. The Explosion test case can be viewed as analogous to the 1D shock tube problem in multidimensions. **Figure 4.27** shows the cross section of the explosion at T=0.25 along the plane $z = 1$.

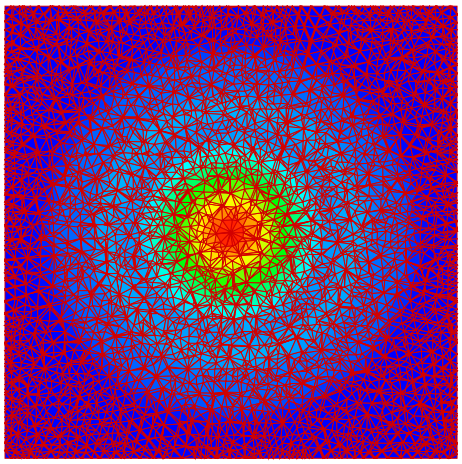
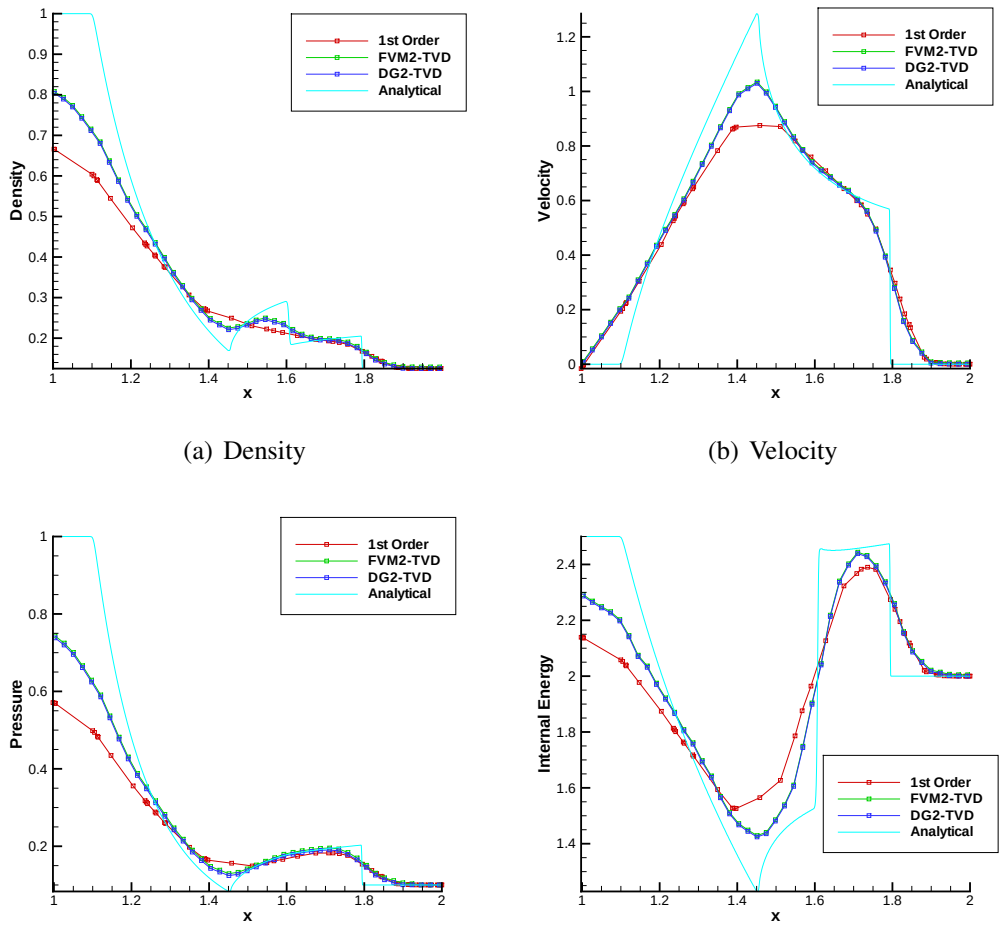


Figure 4.27: Cross-Section of the 40x40x40 Mesh at T=0.25 for DG2-TVD



(a) Density (b) Velocity

(c) Pressure (d) Internal Energy

Figure 4.28: Radial distribution of the explosion test case at T=0.25 for FVM2-TVD and DG2-TVD

The radial distribution of the explosion, taken at the plane $y = 1, z = 1$ prorogating in the x-direction shows comparison between the FVM and DG methods for the 2nd order methods with a TVD limiter, this is also compared with 1st order method, and shows a marked improvement in the resolution(**Figure 4.28**). The degradation of the smooth region of the radial solution in, notably, the centre of the density, pressure and internal energy is a result of a standing wave type phenomena, whereby the resolution of the solution is effected by the TVD limiter. A similar radial solution profile was attained by[67], albeit with a far superior ADER scheme with a courser mesh.

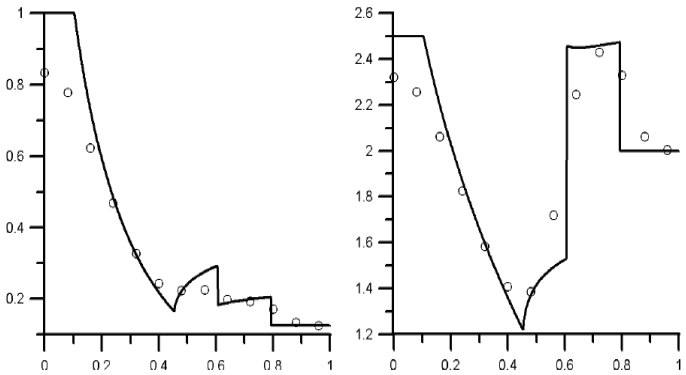


Figure 4.29: ADER- 3rd order for 25x25x25

The spherical shockwave which is a characteristic feature of the explosion testcase is also visible in the Density and Internal energy profile(**Figure 4.30**, **Figure 4.31**) which is prorogating in all directions as would be expected.

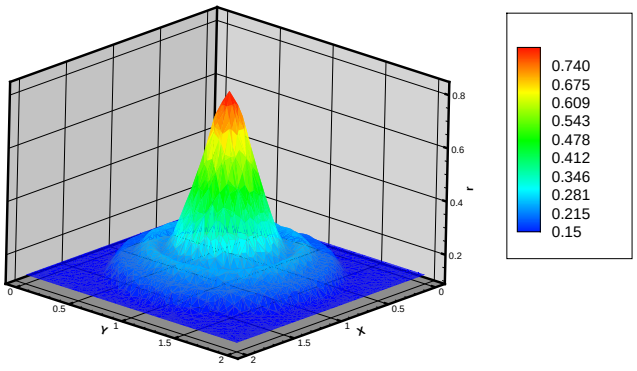


Figure 4.30: Density profile for the Explosion at T=0.25 for DG2-TVD

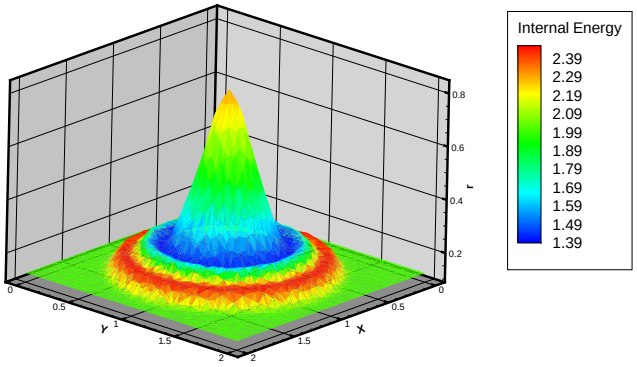


Figure 4.31: Internal Energy profile for the Explosion at T=0.25 for DG2-TVD

The convergence study was not conducted for this test case, as it was envisaged the results would not yield any meaningful insight into the convergence of either methods. The number of grid points and nodes needs to be increased so that the resolution is improved and the convergence can be assumed to be in the asymptotic region. In[69] the same 2nd Order FVM-TVD scheme was applied to the explosion problem, and a marked improvement in the resolution is noticeable(**Figure 4.32**); the spherical shock wave becomes more obvious.

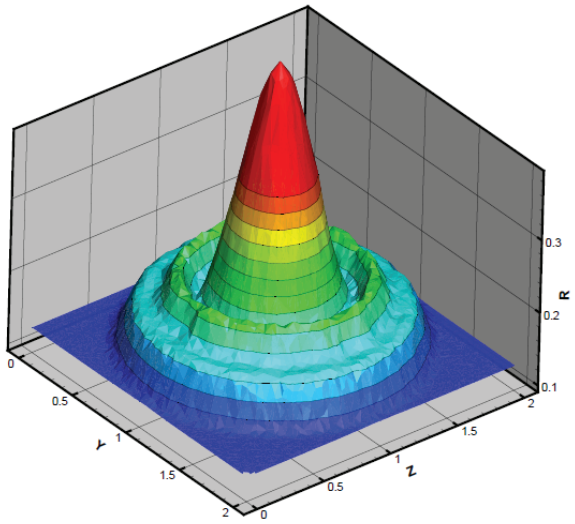


Figure 4.32: TVD-2 for 1.5 million cells [69]

4.6 Conclusion of the testcases

4.6.1 Smooth solutions

Linear Advection equation:
The linear advection equation was solved on four grids for three Initial conditions for the FVM and DG methods for 2nd and 3rd order spatial accuracy for T=1 physical time. For IC=$sin(2\pi x)sin(2\pi y)sin(2\pi z)$ both FVM and DG converged to the nominal order accuracy with the DG method producing lower L_0 and L_1 errors. The DG method was produced error which was 1.7 - 4 times less than FVM for the finest mesh. For increasing wavenumber $W_n = 4, 6$ i.e for the initial conditions $sin(4\pi x)sin(4\pi y)sin(4\pi z)$ and $sin(6\pi x)sin(6\pi y)sin(6\pi z)$ the DG method produced a better resolution than FVM, both methods converged slower to the nominal order of accuracy than they did for the $sin(2\pi x)sin(2\pi y)sin(2\pi z)$ initial condition, with the convergence becoming slower for increasing wavenumber. However, the DG method was effected less by the increasing wavenumber. For example for the initial conditions $sin(6\pi x)sin(6\pi y)sin(6\pi z)$ the DG method was able to produce a very good resolution, as demonstrated by the comparison of the prorogation of the $sin(6\pi x)$ wave in the x direction, and the L_1 error for DG was 8 times less than FVM for the finest grid. With the inclusion of the Barth-Jespersen TVD limiter, the resolution and convergence of both methods was severely effected, with neither FVM or DG attaining the nominal order of accuracy, as was expected. However, the DG method produced a lower error and better resolution than FVM.

Euler equations:
The Euler equations were solved using a method of manufactured solutions, which yields an analytical solutions for purposes for validation, on four grids for 2nd and 3rd order spatial accuracy for T=1 physical time. Similar to the linear advection equation this testcase for the Euler equations produces a similar comparison between FVM and DG with the DG method producing lower error for both L_0 and L_1 errors. With the inclusion of the Barth-Jespersen TVD limiter the convergence to the nominal order of accuracy is degraded.

4.6.2 Discontinuous solutions

Sod's shocktube:
The Sod's shocktube case was solved on four grids for 2nd order accuracy with the inclusion of a Barth-Jespersen limiter. Using the FVM and DG methods without a limiter resulted in a unstable scheme producing negative density and causing the simulation to terminate. The TVD limiter we use in this thesis is only applicable to 2nd order spatial accuracy. Firstly the DG2 method with TVD limiter was compared to the 1st order accurate scheme; the DG2-TVD showed an improvement in the resolution. Although as proved by [22] TVD schemes are at most 1st order accurate in multidimensions, hence the lack of massive improvement in resolution. To verify the convergence of the

DG2-TVD, a comparison was sought with the FVM2-TVD scheme. This comparison investigated the ability of both method to handle the wave solutions of the Riemann problem. Both methods demonstrated relatively good approximation with respect to the analytical solution. The DG method was slightly better: it produced a smaller L_0 and L_1 error and converged faster, however, the convergence to the nominal order of accuracy was not attained.

Implosion testcase: The implosion testcase was conducted as a verfication problem due to the demanding nature of the physical solution of the problem. The test was allowed to run for T=0.4, which is sufficient time for the shockwaves of the implosion to converge to a singularity and then to result in a an subsequent explosion. The DG2-TVD method was robust enough to replicate the implosion testcase. No comparison was sought for the testcase, due to the lack of an analytical solution.

Explosion testcase: The explosion testcase can be viewed as a 3D extension of Sod's shocktube, with the radial solution propogating in all three directions. The explosion testcase was solved on a fine mesh and yield a radial solution for both FVM and DG which was similar. A convergence study was not conducted due to the sub-standard resolution achieved by the TVD methods. This can be overcome by increasing by number of elements as demonstrated by work carried out by another author, although this is not possible on the current computer due to the limitations of the memory.

CHAPTER 5

Conclusion

The Discontinuous Galerkin method for 2nd and 3rd order spatial accuracy (DG2 and DG3) for smooth problems and 2nd order accuracy TVD scheme (DG2-TVD) for problem with discontinuities, along with 2nd and 3rd order temporal accuracy was implemented on the Cranfield University's Finite volume Method solver. The DG method was tested against problems which required no limiter and problems where the use of a limiter is essential to suppress oscillations which may render a numerical scheme unstable.

For the linear advection equation with smooth sine wave type initial conditions, the DG method showed it's superiority, it had a lower L_0, L_1 error than FVM and converged to the nominal order of accuracy. Whereas FVM showed slower convergence when the wavenumber of the sine wave was increased. Similarly the DG scheme for method of manufactured solutions for the Euler equations produced a lower error than FVM.

With the inclusion of the Barth-Jespersen TVD limiter the convergence of both the FVM and DG method was effected. For these so-called discontinuous testcases it was observed that there is very little improvement in the error between the FVM and DG method, and also between 1st order accuracy, this is due to[22]. And similarly there is not an increase in the convergence rate to the nominal order of accuracy between the two methods, as opposed to the smooth solutions problems especially for wavenumbers $W_n = 4, 6$, where we did observe a significant difference in the error and the convergence rate.

The CPU time comparison between the methods revealed that the FVM method is faster than the DG method. The major contributing factor to this is the evaluation of the volume quadratures in the DG method, which is absent in the FVM method.

From a practical prospective i.e the development of an inviscid compressible unsteady solver on an unstructured domain; if one is interested in the development of a new solver, written from 'scratch', then the DG method would serve as an ideal alternative to the FVM solver. The coding aspect of the development is far-less complicated on a unstructured domain for the DG method. For the FVM method, the necessity to increase the stencil as the order of accuracy increases becomes difficult on an unstructured mesh, something which is not required in the DG method. Furthermore, when one is interested in the implementation of a MPI methodology then it is far-more easier

to do for a DG method than the FVM method.

In summary the DG method has proved to more accurate than the FVM method for smooth problems, this is in part due to the greater number of degrees of freedom that DG has over FVM, whereas for problem with shocks, only slightly better than the FVM method. The DG method is approximately eight times slower for a 3rd order method.

5.1 Future work

Having developed the compressible inviscid solver for an unstructured domain composed entirely of tetrahedon elements we can take this work forward and implement the next stage of development.

Hybrid grids

It is possible to extend the 3D DG method to hybrid grids consisting of: tetrahedrons, pyramids, prism and hexhedrons for the initialisation and volume integrals of the algorithm and square and triangles for the evaluation of the surface integrals. This would require the need of specific basis functions corresponding to the shape type, and therefore the need for corresponding quadrature points to solve the integrals. The basis functions are constructed using the same collapsed coordinate method which we used in this thesis. During the collapsed coordinate method, the coordinates of the hexahedron are collapsed to form to a prism, in turn the coordinates of the prism are collapsed to form a pyramid and finally the coordinates of the pyramid are collapsed to form a tetrahdron[59]. The quadrature points for the integrals are constructed in a similar fashion, whereby the gaussian quadrature points required to solve a function on a hexahedron domain is simply the tensor product in the x, y, z directions. Reusing the approach we used to construct the basis functions for different shapes we can construct quadrature points which correspond to these basis functions [33].

Viscous terms

For certain problems the use of an inviscid solver does not produce the desired results. One of the most notable of which (amongst many others) is the flow over a flat plate, where there is a formation of a boundary layer. To extend the solver developed in this thesis to handle the viscous terms we need to extend the work to discretise the Navier-Stoke equations. It is vital, though, that before the Navier-Stokes equations are discretised, that the code has the ability to handle hybrid meshes in particular hexahedron and prism elements. Hexahedron and Prism elements are required within the boundary layer of the flow to accommodate the viscous stress terms which need to be perpendicular to the boundary wall. The initial step in the discretising process requires replace the gradient of the vector of the conserved variables with a vector of unknown

variables Θ thus reducing the original set of equations to a new set of equations which contain only first-order terms. There are many approaches to tackling to the second-order terms and the Navier-Stokes equations, which differ on the choice of the fluxes $\hat{\Theta}$ and the approximate solutions U_h[4, 5, 11]. Each of one of these methods have particular drawbacks and advantages which is discussed in the work by Arnold et.al [1]

WENO limiters

A TVD scheme is widely known to be very dissipative, and for problems which require a large amount of time to develop can be impractical. To overcome this problem the class of WENO methods developed for the FDM and FVM method have been extended to the DG method. Unfortunately, the WENO method for DG no longer preserves the fixed-stencil as the order of accuracy increases, which is something that has been argued as positive feature of the method. However, the results accumulated from the catalogue of work coupling DG-WENO are more superior than those achieved from the DG-TVD methods.

Bibliography

[1] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal*, pages 1749–1779, 2001.

[2] T. Barth and H. Deconinck. *High-Order Methods for Computational Physics*. Springer, 1999.

[3] T.J. Barth and D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA Paper*, 89(89-0366), 1989.

[4] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 131(2):267–279, 1997.

[5] C.E. Baumann and J.T. Oden. A discontinuous hp finite element method for convection-diffusion problems. *Comput.Methods Appl.Mech Engrg*, 175:311–341, 1999.

[6] A.N. Brooks and T.J.R. Hughes. Streamline Upwind/Petrov-Galerkin Formulation for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations. *Computer in Applied Mechanics and Engineering*, 32:199–259, 1982.

[7] M. Cada and M. Torrilhon. Compact third-order limiter functions for finite volume methods. *Journal of Computational Physics*, 228:4118–4145, 2009.

[8] G. Chavent and G. Salzano. A finite element method for the 1D water flooding problem with gravity. *Journal of Computational Physics*, 45:307–344, 1982.

[9] B. Cockburn, B. Lin, and C-W. Shu. TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws III: One-Dimensional Systems. *Journal of computational Physics*, 84:90–113, 1989.

[10] B. Cockburn and C-W. Shu. TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws II: General Framework. *JSTOR Mathematics of Computation*, 52:411–435, April 1989.

[11] B. Cockburn and C-W. Shu. The Local Discontinuous Galerkin Method For Time-Dependent Convection-Diffusion Systems, 1998.

[12] B. Cockburn and C-W. Shu. The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems. *Journal of Computational Physics*, 144:199–224, 1998.

[13] B. Cockburn and C-W. Shu. Runge-Kutta Discontinuous Galerkin Mehtods for Convection-Dominated Problems. *Journal of Scientific Computing*, 16(3):173–261, September 2001.

[14] G. Cohen, X. Ferrieres, and S. Pernet. A Spatial High-Order Hexahedral Discontinuous Method to solve Maxwell's Equations in time domain. *Journal of Computational Physics*, 217(2):340–363, September 2006.

[15] P. Colella and P.R. Woodward. The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations. *Journal of Computational Physics*, 54:174–201, 1984.

[16] S.S. Collis. Discontinuous Galerkin methods for Turbulence Simulation. In *Proceedings of the Summer Program, Center of Turbulence Research*, pages 155–167, 2002.

[17] D. Drikakis and W. Rider. *High-Resolution Methods for Incompressible and Low Speed Flows*. Springer, 2005.

[18] M Dumbser and M Käser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *J. Comput. Physics*, 221(2):693–723, 2007.

[19] D.A. Dunavant. High degree efficent symmetrical Gaussian quadrature rules for the triangle. *International journal for numerical methods in engineering*, 21:1129–1148, 1985.

[20] C. Eskilsson and S.J. Sherwin. A triangular spectral/hp discontinuous Galerkin method for modelling 2D shallow water equations. *Int. J. Numer. Meth. Fluids*, 45:605–623, 2004.

[21] J.E. Flaherty, L. Krivodonova, J-F. Remacle, and M.S. Shephard. Aspects of discontinuous Galerkin methods for Hyperbolic conservation laws. *Finite Elements in Analysis and Design*, 38:889–908, 2002.

[22] J.B. Goodman and R.J. LeVeque. On the Accuracy of Stable Solutions for 2D Scalar Conservation Laws. *Mathematics of Computation*, 45:15–21, 1985.

[23] D. Gottlieb and C-W. Shu. On the Gibbs Phenomenon and its resolution . *SIAM Rev*, 39(4):644–668, 1997.

[24] Sigal Gottlieb and Chi wang Shu. Total variation diminishing runge-kutta schemes. *Math. Comp*, 67:73–85, 1998.

[25] P.C. Hammer and A.H. Stroud. Numerical integration over simplexes and cones. *Mathematical Tables and other aids to computation*, 10:130–137, 1956.

[26] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 59:357–393, 1983.

[27] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly High Order Accurate Essentially Non-Oscillatory Schemes III. *Journal of Computational Physics*, 71:231–303, 1987.

[28] J.S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods*. Springer, 2008.

[29] M. Iskandarani, J. C. Levin, B. J. Choi, and D. B. Haidvogel. Comparison of advection schemes for high-order h-p finite element and finite volume methods. *Ocean Modelling*, 10(1-2 SPEC. ISS.):233–252, 2005.

[30] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. *Convergence*, (81-1259):1–19, 1981.

[31] Y. Jinyun. Symmetric Gaussian quadrature formulae for tetrahedral regions. *Computer Methods in Applied Mechanics and Engineering*, 43:349–353, 1984.

[32] E. Johnsen and T. Colonius. Implementation of WENO schemes in compressible multicomponent flow problems. *Journal of Computational Physics*, 219(2):715–732, 2006.

[33] G. Karniadakis and S.J. Sherwin. *Spectral/hp Element Methods for CFD*. Oxford University Press, 1999.

[34] P. Keast. Moderate-degree tetrahedral quadrature formulas. *Computer Methods in Applied Mechanics and Engineering*, 55:339–348, 1986.

[35] D. Kim and D. Choi. A Second-Order Time-Accurate Finite Volume Method for Unsteady Incompressible Flow on Hybrid Unstructured Grids. *Journal of Computational Physics*, 162(2):411 – 428, "2000.

[36] C.M. Klaij, J.J.W. van der Vegt, and H. van der Ven. Space-Time Discontinuous Galerkin Method for the Compressible Navier-Stokes equations. *Journal of Computational Physics*, 217(2):589–611, September 2006.

[37] N.E. Kolgan. Application of the minimum-derivative principle in the construction of finite-difference schemes for numerical analysis of discontinuous solutions in gas dynamics. *Uch.Zap.TsaGI*, 3(6):68–77, 1972.

[38] C.B. Laney. *Computational Gasdynamics*. Cambridge University Press, 1998.

[39] B.V. Leer. Towards the Ultimate Conservative Difference Scheme. V. A Second Order Sequal to Godunov's Method . *Journal of Computational Physics*, 32:101–136, 1979.

[40] B.V. Leer. Upwind and High-Resolution Methods for Compressible Flow: From Donor Cell to Residual-Distribution Schemes . *COMMUNICATIONS IN COMPUTATIONAL PHYSICS*, 1(2):192–206, 1997.

[41] B.Q. Li. *Discontinuous Finite Elements in Fluid Dynamics and Heat Transfer*. Springer, 2006.

[42] H. Liu and J. Yan. A Local Discontinuous Galerkin Method for the Kortewag-de Vries with for Boundary effect. *Journal of Computational Physics*, 215:197–218, 2006.

[43] Y. Liu, M. Vinokur, and Z.J. Wang. Spectral Difference Method for Unstructured grids I: Basic formulation. *Journal of Computational Physics*, 216:780–801, 2006.

[44] I. Lomtev and G. Karniadakis. A Discontinuous Method for the Navier-Stokes Equations. *International Journal for Numerical Methods In Fluids*, 152(2):608–641, July 1999.

[45] I. Lomtev, R.M. Kirby, and G.E. Karniadakis. A Discontinuous Galerkin ALE method for compressible viscous flows in moving domains. *J.Comp.Phys*, 155(1):128–159, 1999.

[46] X-D. Lui, S. Osher, and T. Chan. Weighted Essentially Non-Oscillatory Schemes. *Journal of Computational Physics*, 115, 1994.

[47] H. Luo, J.D. Baum, and R. Lohner. A Hermite Weno-based Limiter for Discontinuous Galerkin Method on Unstructured Grids. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, volume AIAA 2007-510. AIAA, January 2007.

[48] K. Michalak and C. Ollivier-Gooch. Limiters for unstructured higher-order accurate solutions of the euler equations. *AIAA Forty-Sixth Aerospace Sciences Meeting*, 2008.

[49] R.C. Millington, V. Titarev, and E.F. Toro. ADER: Arbitary-Order Non-Oscillatory Advection Schemes. In Heinrich Freistühler and Gerald Warnecke, editors, *Hyperbolic Problems: theory, numerics, applications: eighth international conference in Magdeburg, February, March 2000*, volume 141 of *International series of numerical mathematics*. Birkhäuser, Basel, 2001.

[50] K.V. Nagaraja and H.T. Rathod. Symmetric Gauss Legendre Quadrature Rules for Numerical Integration over an Arbitrary Linear Tetrahedra in Euclidean Three-Dimensional Space. *Int.Journal of Math. Analysis*, 4:921–928, 2010.

[51] X. Nogueira, L. Cueto-Felgueroso, I. Colominas, H. Gmez, F. Navarrina, and M. Casteleiro. On the accuracy of finite volume and discontinuous galerkin discretizations for compressible flow on unstructured grids. *International Journal for Numerical Methods in Engineering*, 78(13):1553–1584, 2009.

[52] C Ollivier-Gooch and M Van Altena. A High-Order-Accurate Unstructured Mesh Finite-Volume Scheme for the AdvectionDiffusion Equation. *Journal of Computational Physics*, 2002.

[53] T.H. Pulliam and J.L Steger. Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow. *AIAA*, 18(2):159–167, February 1980.

[54] J. Qiu and C-W. Shu. Runge-Kutta Discontinuous Galerkin Method using Weno Limiters. *SIAM Journal of Scientific Computing*, 26(3):907–929, 2005.

[55] J. Qui, M. Dumbser, and C-W. Shu. The discontinuous galerkin with lax-wendroff type time discretizations. *Computer methods in applied mechanics and engineering*, 194:4528–4543, 2005.

[56] H.T. Rathod, K.V. Nagaraja, B. Venkatesudu, and L. Ramesh. Gauss Legendre quadrature over a triangle. *J. Indian.Inst.Sci*, 84:183–188, 2004.

[57] W.H. Reed and T.R. Hill. Triangular Mesh Methods for the Neutron Transport Equation. *Proceedings of the American Nuclear society*, 1973.

[58] M. Remaki and W.G. Habashi. A discontinuous Galerkin method/HLLC solver for the Euler equations. *Int. J. Numer. Meth. Fluids*, 43:1391–1405, 2003.

[59] S.J. Sherwin. Hierarchical hp finite elements in hybrid domains. *Finite Elements in Analysis and Design*, 27:109–119, 1997.

[60] C-W. Shu. High-Order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD. *International Journal of Computational Fluid Dynamics*, 17(2):107–118, 2003.

[61] C-W. Shu and S. Osher. Effcient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes. *Journal of Computational Physics*, 77(2):439–471, August 1988.

[62] G.A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27:1–31, 1978.

[63] A. Suresh and H.T. Huynh. Accurate monotoncity-perserving schemes with Runge-Kutta time stepping. *Journal of Computational Physics*, 136:83–99, 1997.

[64] P.K. Sweby. High-resolution schemes using flux limiters for hyperbolic conservation-laws. *SIAM J.NUMER. ANALYSIS*, 1984.

[65] R.I. Tanner and S-C. Xue. Computing Transient Flows with High Elasticity. *Korea-Australia Rheology*, 14(4):143–159, December 2002.

[66] H. Ten Eyck and A. Lew. Discontinuous Galerkin for NonLinear Elasticity. In *III European Conference on Computational Mechanics Solids, Structures and Coupled Problems in Engineering*, 2006.

[67] V. Titarev. *Derivative Riemann problem and ADER schemes*. PhD thesis, University of Trento.

[68] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 2009.

[69] P. Tsoutsanis. *Very high-order methods for arbitrary unstructured grids*. PhD thesis, Cranfield University.

[70] S. Tu and S. Aliabadi. A Slope Limiting Procedure in Discontinuous Galerkin Finite Element Method for Gasdynamics Applications. *International Journal of Numerical Analysis and Modelling*, 2(2):163–178, 2005.

[71] T. Warburton and G. Karniadakis. A Discontinuous Method for the Viscous MHD Equations. *Journal of Computational Physics*, 152(2):608–641, July 1999.

[72] J.F. Wendt. *Computational Fluid Dynamics*. Springer, 1996.

[73] K. Yasue, M. Furudate, N. Ohnish, and K. Sawada. Implicit Discontinuous Galerkin Method for RANS Simulation Utilizing Pointwise Relaxation Algorithm. *Commun.Comput.Phys.*, 7:3:510–533, 2010.

[74] M. Zhang and C. Shu. An analysis of and a comparison between the discontinuous galerkin and the spectral finite volume methods. *Computers and Fluids*, 34(4-5 SPEC.ISS.):581–592, 2005.

[75] T. Zhou, Y. Li, and C. Shu. Numerical comparison of weno finite volume and runge-kutta discontinuous galerkin methods. *Journal of Scientific Computing*, 16(2):145–171, 2001.

[76] J. Zhu, J. Qiu, C-W. Shu, and M. Dumbser. Runge-Kutta Discontinuous Galerkin Method using WENO Limiters II: Unstructured Meshes. *Journal of Computational Phyics*, 227:4330–4353, 2008.

[77] O.C. Zienkiewicz. *The Finite Element Method*. McGraw-Hill, 1977.

[78] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method for Solid and Structural Mechanics*. Elsevier, 2005.

[79] O.C. Zienkiewicz, R.L. Taylor, S.J. Sherwin, and Peiro.J. On Discontinuous Galerkin Methods. *International journal for numerical methods in engineering*, 58(8):1119–1148, October 2003.

Appendices

CHAPTER A

Euler Equations

The compressible Euler equations describe the motion of fluids with variable density, without the effects of viscosity, heat conduction and body forces. Even with the exclusion of the pre-mentioned physical effects the Euler equations can provide sufficiently reasonable approximations for certain phenomena. For example, a good approximation for the lift generated on an airfoil with a low-angle of attack can be obtained from the Euler equations. However, for certain situations the use of the Euler equations may not provide the desired results, a well known case being the growth of a boundary layer on a thin plate. The boundary layer contains most of the viscosity of the fluid, therefore, not addressing the viscosity in the governing equations leads to inadequate results.

The Euler equations are a system of non-linear hyperbolic equations, and as result have wave-type solutions.

The three dimensional case consists of 6 *primitive variables*

in conservation form

$$\frac{\partial U}{\partial t} + \nabla.\mathbf{F}(\mathbf{U}) = 0$$

or

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial F(\mathbf{U})}{\partial x} + \frac{\partial G(\mathbf{U})}{\partial y} + \frac{\partial H(\mathbf{U})}{\partial z} = 0$$

the vector of conserved variables are

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix}$$

and the conservative fluxes are:

$$F(\mathbf{U}) = \begin{pmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix}, \quad G(\mathbf{U}) = \begin{pmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ \rho vw \\ v(E+p) \end{pmatrix}, \quad H(\mathbf{U}) = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ p + \rho w^2 \\ w(E+p) \end{pmatrix}$$

Variable	Definition
ρ	Density
u	velocity component
v	velocity component
w	velocity component
p	pressure

Table A.1: Primitive variables of the Euler equations

The system consists of five equations and six unknowns, and thus requires closing. This is done using an equation of state, with one of the most common being the ideal gas law

$$p = \rho(\gamma - 1)e$$

and the energy is given by

$$E = \rho(\frac{1}{2}|\mathbf{u}|^2 + e)$$

where γ is the gas constant, and e is the internal energy. The five equations represent the *conservation of mass*, *momentum* and *energy*.

The Euler equations can be written in a quasi-linear form i.e as a system of primitive variables

$$\mathbf{W}_t = \mathbf{A}(\mathbf{W})\mathbf{W}_x + \mathbf{B}(\mathbf{W})\mathbf{W}_y + \mathbf{C}(\mathbf{W})\mathbf{W}_z$$

Where $\mathbf{W}$ is the vector of primitive variable, as defined in table (A.1). The matrices $\mathbf{A}(\mathbf{W})$, $\mathbf{B}(\mathbf{W})$ and $\mathbf{C}(\mathbf{W})$ are defined as follows:

$$\mathbf{A}(\mathbf{W}) = \begin{pmatrix} u & \rho & 0 & 0 & 0 \\ 0 & u & 0 & 0 & 1/\rho \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & \rho a^2 & 0 & 0 & u \end{pmatrix}$$

$$\mathbf{B}(\mathbf{W}) = \begin{pmatrix} v & \rho & 0 & 0 & 0 \\ 0 & v & 0 & 0 & 0 \\ 0 & 0 & v & 0 & 1/\rho \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & \rho a^2 & 0 & v \end{pmatrix}$$

$$\mathbf{C}(\mathbf{W}) = \begin{pmatrix} w & \rho & 0 & 0 & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w & 1/\rho \\ 0 & 0 & 0 & \rho a^2 & w \end{pmatrix}$$

The primitive form of the Euler equations are quite useful in the analysis of the Riemann problem, in particular we can establish the eigenvalues and eigenvectors from the primitive form.

CHAPTER **B**

Basis Functions

The Basis functions are based upon the orthogonal *Jacobi polynomials*. The basis functions are defined in the reference space, and based upon the concept of the collapsed coordinate method

where the Jacobi polynomials are defined as

$$\begin{aligned}
\psi_p^a(z) &= P_p^{0,0}(z) \\
\psi_{pq}^b(z) &= \left(\frac{1-z}{2}\right)^p P_q^{2p+1,0}(z) \\
\psi_{pqr}^c(z) &= \left(\frac{1-z}{2}\right)^{p+q} P_r^{2p+2q+2,0}(z)
\end{aligned}$$

the basis functions for the tetrahedron are defined using the principle functions as

$$\phi_l(\xi,\eta,\zeta) = \psi_p^a\left(\frac{2\xi}{1-\eta-\zeta}-1\right)\psi_{pq}^b\left(\frac{2\eta}{1-\zeta}-1\right)\psi_{pqr}^c\left(-1+2\zeta\right)$$

Basis Functions	
l	ϕ_l
1	1
2	$2\xi+\eta+\zeta-1$
3	$-1+\zeta+3\eta$
4	$-1+4\zeta$

Table B.1: Basis functions

l	$\frac{\partial\phi_l}{\partial\xi}$	$\frac{\partial\phi_l}{\partial\eta}$	$\frac{\partial\phi_l}{\partial\zeta}$
1	0	0	0
2	2	1	1
3	0	3	1
4	0	0	4

Table B.2: Partial derivatives of the Basis functions

From the basis functions it is possible to derive the norms. The basis functions are orthogonal ,therefore, the mass matrix $\mathbf{M}^k$ are diagonal, and the norms are the diagonal in the mass matrix. The mass matrix is given for upto 4^{th} order accuracy.

The mass matrix for the P^0 (1^{st} order case) is

$$\mathbf{M}^0 = \begin{pmatrix} \frac{1}{6} \end{pmatrix}$$

The mass matrix for the P^1 (2^{nd} order case) is

$$\mathbf{M}^1 = \begin{pmatrix} \frac{1}{6} & 0 & 0 & 0 \\ 0 & \frac{1}{60} & 0 & 0 \\ 0 & 0 & \frac{1}{20} & 0 \\ 0 & 0 & 0 & \frac{1}{10} \end{pmatrix}$$

CHAPTER C

Derivation of the metrics

The metrics are used in the derivation of the volume quadratures, in particular when the integral is mapped to the reference space.

$$\begin{aligned} x^P &= x_1 + (x_2 - x_1)\xi^p + (x_3 - x_1)\eta^p + (x_4 - x_1)\zeta^p \\ y^P &= y_1 + (y_2 - y_1)\xi^p + (y_3 - y_1)\eta^p + (y_4 - y_1)\zeta^p \\ z^P &= z_1 + (z_2 - z_1)\xi^p + (z_3 - z_1)\eta^p + (z_4 - z_1)\zeta^p \end{aligned} \tag{C.0.1}$$

the partial derivatives notation will be defined as follows (for clarity we drop the p superscript),

$$\frac{\partial x}{\partial \xi} = x_\xi, \qquad \frac{\partial x}{\partial \eta} = x_\eta, \qquad \text{etc.} \tag{C.0.2}$$

for the volume quadratures we want to establish the relationships

$$\frac{\partial \xi}{\partial x} = \xi_x, \qquad \frac{\partial \xi}{\partial y} = \xi_y, \qquad \text{etc.} \tag{C.0.3}$$

to do this, we exploit the following relationship between the matrices

$$\begin{pmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{pmatrix} \begin{pmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

we know the left matrix, which can be established from taking the partial derivatives of (C.0.1). We invert this matrix to get

$$\begin{pmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{pmatrix} = \begin{pmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Again, for clarity we can remove the identity matrix to get

$$\begin{pmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{pmatrix} = \begin{pmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{pmatrix}^{-1}$$

Finally to establish the partial derivatives (C.0.3) we invert the 3×3 matrix of the partial derivatives of (C.0.2).

$$
\begin{aligned}
\xi_x &= \frac{1}{J}(y_\eta z_\zeta - z_\eta y_\zeta) \\
\xi_y &= -\frac{1}{J}(x_\eta z_\zeta - z_\eta x_\zeta) \\
\xi_z &= \frac{1}{J}(x_\eta y_\zeta - y_\eta x_\zeta) \\
\eta_x &= -\frac{1}{J}(y_\xi z_\zeta - z_\xi y_\zeta) \\
\eta_y &= \frac{1}{J}(x_\xi z_\zeta - z_\xi x_\xi) \\
\eta_z &= -\frac{1}{J}(x_\xi y_\zeta - y_\xi x_\zeta) \\
\zeta_x &= \frac{1}{J}(y_\xi z_\eta - y_\eta z_\xi) \\
\zeta_y &= -\frac{1}{J}(x_\xi z_\eta - x_\eta z_\xi) \\
\zeta_z &= \frac{1}{J}(x_\xi y_\eta - x_\eta y_\xi)
\end{aligned}
$$

the term J is the Jacobian given by the following expression

$$J = x_\xi(y_\eta z_\zeta - z_\eta y_\zeta) - y_\xi(x_\eta z_\zeta - z_\eta x_\zeta) + z_\xi(x_\eta y_\zeta - y_\eta x_\zeta)$$

CHAPTER D

Runge-Kutta Methods

A m^{th} order Explicit Runge-Kutta method is given by

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t \sum_{i=1}^{M} c_i k_i$$

where

$$\begin{aligned} k_1 &= f(t_n, \mathbf{x}_n) \\ k_2 &= f(t_n + \alpha_2 h, \mathbf{x}_n + \Delta t \beta_{21} k_1(t_n, \mathbf{x}_n)) \\ k_2 &= f(t_n + \alpha_3 h, \mathbf{x}_n + \Delta t(\beta_{31} k_1(t_n, \mathbf{x}_n) + \beta_{32} k_2(t_n, \mathbf{x}_n)) \\ &\vdots \\ k_m &= f(t_n + \alpha_m h, \mathbf{x}_n + \Delta t \sum_{j=1}^{m-1} \beta_{mj} k_j) \end{aligned} \tag{D.0.1}$$

where Δt is the time interval $t_{n-1} - t_n$

the Runge-Kutta coefficients $c_i, \alpha_m, \beta_{mj}$ can be summarised in the Butcher's tableau table: D.1

Table D.1: General form of the Butcher tableau

0						
α_2	β_{21}					
α_3	β_{31}	β_{32}	$\ddots$			
$\vdots$	$\vdots$	$\vdots$		$\ddots$		
$\vdots$	$\vdots$	$\vdots$				
α_m	β_{m1}	β_{m2}	$\dots$	$\dots$	β_{mm-1}	
	c_1	c_2	$\dots$	$\dots$	c_{m-1}	c_m

The Butcher tableau for the coefficients of the non-TVD Runge-Kutta method are given below for up to 6th order accuracy(tables: D.2 - D.7).

Table D.2: Butcher tableau for 1st order Runge-Kutta (Euler Method)

0	
	1

Table D.3: Butcher tableau for 2nd order Runge-Kutta

0		
1/2	1/2	
	0	1

Table D.4: Butcher tableau for 3rd order Runge-Kutta

0	0		
1/2	1/2		
1	-1	0	1
	1/6	2/3	1/6

Table D.5: Butcher tableau for 4th order Runge-Kutta

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

Table D.6: Butcher tableau for 5th order Runge-Kutta

0						
1/4	1/4					
1/4	1/8	1/8				
1/2	0	-1/2	1			
3/4	3/16	0	0	9/16		
1	-3/7	2/7	12/7	-12/7	8/7	
	7/90	0	32/90	12/90	32/90	7/90

Table D.7: Butcher tableau for 6th order Runge-Kutta

0							
1/3	1/3						
2/3	0	2/3					
1/3	1/12	3	-1/12				
1/2	-1/16	18/16	-3/16	-6/16			
1/2	0	9/8	-3/8	-6/8	1/2		
1	9/44	-36/44	63/44	72/44	64/44	0	
	11/120	0	81/120	81/120	-32/120	-32/120	11/120

The TVD Runge-Kutta methods for 2nd and 3rd order accuracy are given below (equations D, D), these are given in their full form, because tabling the method in a Butcher tableau is not possible. The $\mathbf{x}_n$ at each stage of the non-TVD Runge-Kutta method is unitary, however, in the TVD Runge-Kutta method a coefficient has to multiple the $\mathbf{x}_n$. Where $\mathbf{L_h}$ is the right-hand side of the ODE.

$$\begin{aligned} u^1 &= u^0 + \Delta t \mathbf{L_h}(u^0) \\ u^{n+1} &= \frac{1}{2}u^0 + \frac{1}{2}u^1 + \frac{1}{2}\Delta t \mathbf{L_h}(u^1) \end{aligned} \tag{D.0.2}$$

$$\begin{aligned} u^1 &= u^0 + \Delta t \mathbf{L_h}(u^0) \\ u^2 &= \frac{3}{4}u^0 + \frac{1}{4}u^1 + \frac{1}{4}\Delta t \mathbf{L_h}(u^1) \\ u^{n+1} &= \frac{1}{3}u^0 + \frac{2}{3}u^1 + \frac{2}{3}\Delta t \mathbf{L_h}(u^2) \end{aligned} \tag{D.0.3}$$